



# **GRADO EN INGENIERÍA INFORMÁTICA**

## **TRABAJO DE FINAL DE GRADO**

---

### **Aplicación web para la gestión de bibliotecas para Centros Sociales (BiblioCloud)**

---

*Autor:*

Carlos BACETE MILLÁN

*Supervisor:*

José Manuel TIRADO FELIP

*Tutor académico:*

M. Ángeles LÓPEZ MALO

**Fecha de lectura: 13 de julio de 2021**

Curso académico 2020/2021

## Resumen

El presente documento contiene la memoria del Trabajo Final de Grado de Ingeniería Informática, donde se expone el desarrollo de una aplicación web para la gestión de una biblioteca realizada durante la estancia en prácticas en la empresa Inttegrum. La aplicación va dirigida a los trabajadores de las bibliotecas que hagan uso de ella.

El sistema ha sido desarrollado de tal forma que se puedan gestionar todos los aspectos necesarios de una biblioteca. Esto incluye la gestión de libros, la gestión de préstamos y la gestión de incidencias. Además, la aplicación también incluye un sistema para generar estadísticas.

El proyecto se ha desarrollado siguiendo una metodología predictiva tradicional, donde desde un principio se programaron una serie de tareas que se han ido completando en el tiempo establecido.

La aplicación se ha desarrollado utilizando varios lenguajes de programación. En concreto se ha utilizado el *framework* Yii2, Java, HTML, CSS y MySQL. También se hace uso de un servidor en Plesk, donde está almacenada la base de datos y desde donde se lanza la aplicación.

## Palabras clave

Biblioteca, libros, préstamos, incidencias, accesos, estadísticas, Yii2, BiblioCloud.

## Keywords

Library, books, loans, incidents, access, statistics, Yii2, BiblioCloud.

# Índice general

<b>1. Introducción</b>	<b>9</b>
1.1 Contexto y motivación del proyecto . . . . .	9
1.2 Descripción del proyecto . . . . .	10
1.3 Objetivos del proyecto . . . . .	11
1.4 Alcance del proyecto . . . . .	12
1.4.1 Alcance funcional . . . . .	12
1.4.2 Alcance organizativo . . . . .	13
1.4.3 Alcance informático . . . . .	13
1.5 Estructura de la memoria . . . . .	14
<b>2. Planificación y seguimiento del proyecto</b>	<b>15</b>
2.1 Metodología . . . . .	15
2.2 Tecnologías y herramientas . . . . .	16
2.3 Planificación temporal . . . . .	17
2.3.1 Fases y tareas . . . . .	17
2.3.2 Estructura de Desglose de Trabajo . . . . .	18
2.3.3 Distribución temporal . . . . .	19
2.3.4 Diagrama de Gantt . . . . .	19
2.4 Estimación de recursos y costes del proyecto . . . . .	22
2.4.1 Costes hardware . . . . .	22
2.4.2 Costes software . . . . .	23
2.4.3 Costes humanos . . . . .	23
2.5 Gestión de riesgos . . . . .	25
2.6 Seguimiento del proyecto . . . . .	26
2.6.1 Inicio del proyecto y fase de comunicación . . . . .	26
2.6.2 Desarrollo de la implementación . . . . .	26
2.6.3 Etapa final . . . . .	27

<b>3. Análisis del sistema</b>	<b>29</b>
3.1 Diagrama de casos de uso . . . . .	29
3.2 Requisitos funcionales . . . . .	30
3.3 Diagrama de clases . . . . .	36
<b>4. Diseño del sistema</b>	<b>39</b>
4.1 Diseño de la arquitectura del sistema . . . . .	39
4.2 Diseño de la base de datos . . . . .	40
4.2.1 Población previa de la base de datos . . . . .	40
4.3 Diseño de las interfaces de usuario . . . . .	40
<b>5. Implementación</b>	<b>43</b>
5.1 Detalles de la implementación . . . . .	43
5.2 Desarrollo y resultados . . . . .	44
5.2.1 Menú principal . . . . .	46
5.2.2 Gestión de libros . . . . .	49
5.2.3 Gestión de préstamos e incidencias . . . . .	51
5.2.4 Gestión de clientes . . . . .	53
5.2.5 Gestión de accesos . . . . .	54
5.2.6 Generación de estadísticas . . . . .	55
<b>6. Pruebas y validación</b>	<b>59</b>
6.1 Pruebas unitarias y de integración . . . . .	59
6.2 Validación . . . . .	59
<b>7. Conclusiones</b>	<b>61</b>
7.1 Conclusiones técnicas . . . . .	61
7.2 Conclusiones personales . . . . .	61
<b>Bibliografía . . . . .</b>	<b>63</b>

# Índice de figuras

1.1 Logos de Inttegrum [1] y Mediterránea Gestión [2] . . . . .	9
2.1 Esquema de la metodología en cascada . . . . .	15
2.2 Lenguajes de programación empleados . . . . .	16
2.3 EDT del proyecto . . . . .	18
2.4 Distribución temporal de las tareas . . . . .	20
2.5 Diagrama de Gantt . . . . .	21
3.1 Diagrama de casos de uso . . . . .	29
3.2 Diagrama de clases . . . . .	37
4.1 Patrón de diseño MVC. Fuente: Seul Lee [15] . . . . .	39
4.2 Vista índice con <i>grid</i> . . . . .	41
4.3 Iconos de la columna de acciones . . . . .	41
4.4 Botones Bootstrap . . . . .	41
5.1 Paquetes de controladores y vistas . . . . .	43
5.2 Paquetes de modelos y búsqueda . . . . .	44
5.3 Formulario . . . . .	45
5.4 PDF generado con los datos del modelo seleccionado . . . . .	46
5.5 Menú principal . . . . .	47
5.6 Barra lateral . . . . .	48
5.7 Búsqueda de libro . . . . .	48
5.8 <i>Grid</i> de libros . . . . .	49
5.9 Formulario para dar de alta un libro . . . . .	50
5.10 Campo ISBN del formulario de libros . . . . .	51
5.11 Botones que despliegan modals . . . . .	51
5.12 Vista del libro disponible . . . . .	52
5.13 Vista del libro en préstamo . . . . .	53

5.14	Vista del libro para el cliente . . . . .	54
5.15	Control accesos . . . . .	54
5.16	Estadísticas de accesos . . . . .	55
5.17	Gráficas generadas . . . . .	56
5.18	Estadísticas de préstamos . . . . .	56
5.19	Ranking . . . . .	57

# Índice de tablas

2.1	Costes hardware . . . . .	22
2.2	Costes software . . . . .	23
2.3	Costes humanos . . . . .	24
2.4	Costes indirectos . . . . .	24
2.5	Coste total del proyecto . . . . .	25
3.1	Requisito funcional CU01 . . . . .	30
3.2	Requisito funcional CU02 . . . . .	31
3.3	Requisito funcional CU03 . . . . .	32
3.4	Requisito funcional CU04 . . . . .	33
3.5	Requisito funcional CU05 . . . . .	34
3.6	Requisito funcional CU06 . . . . .	35
3.7	Requisito funcional CU07 . . . . .	36





# Capítulo 1

## Introducción

### 1.1 Contexto y motivación del proyecto

El proyecto documentado en esta memoria se desarrolló durante la estancia de prácticas en la empresa Inttegrum Consultora y Gestora Del Mediterraneo SL, una empresa pequeña formada por tres empleados que se dedica principalmente al desarrollo de webs y aplicaciones de gestión, aunque también dispone de servicio técnico y cursos informáticos [1]. Inttegrum colabora con Mediterránea Gestión, una compañía que trata de facilitar la vida de las personas tanto en el ámbito educativo como en el social [2]. Para ello dispone de distintos programas que ayudan a conseguir este propósito, los cuales se utilizan tanto en centros de mayores como en escuelas infantiles.



Figura 1.1 : Logos de Inttegrum [1] y Mediterránea Gestión [2]

Justo de esta colaboración surge la idea del proyecto. Mediterránea Gestión contacta con Inttegrum para desarrollar una aplicación para gestionar la biblioteca de uno de los centros de mayores que controla, en concreto, el Centro Integral de Mayores (CIM) de Oropesa del Mar [3].

La principal motivación de la aplicación, cuyo nombre es BiblioCloud y es el que usaremos a partir de ahora para referirnos a ella durante todo el proyecto, es facilitar las tareas de gestión de la biblioteca a todos los trabajadores del centro de mayores que se encargan de ella. Estas tareas incluyen tanto la gestión de libros como la gestión de préstamos e incidencias de los mismos.

## 1.2 Descripción del proyecto

Como ya he explicado anteriormente, el proyecto surge debido a la necesidad de actualizar el sistema mediante el cual se gestiona la biblioteca de un centro social que controla la empresa. El sistema que utilizaban estaba muy anticuado, ya que todo se hacía por medio de papeles y documentos. Para poder realizar esta actualización tecnológica, se ha desarrollado BiblioCloud una aplicación web que facilita la realización de las tareas de los responsables de la biblioteca, mediante la cual se podrá llevar un control de los libros actuales y de los préstamos de estos a los clientes, registrados previamente. En los siguientes párrafos se explica el propósito de la aplicación de forma más detallada.

En primer lugar, no todo el mundo tendrá acceso. El uso está restringido a dos tipos de usuarios: los trabajadores del centro social encargados de la biblioteca, que serán los que habitualmente harán uso de ella; y los administradores, la empresa Inttegrum Consultora y Gestora del Mediterraneo, SL.

Los clientes serán registrados por el trabajador la primera vez que quieran realizar un nuevo préstamo. Primero se revisará si existe y, en caso de que no lo esté, se guardarán sus datos (DNI, nombre y apellidos, dirección, teléfono, localidad, etc.). Estos clientes son aquellas personas que acuden a la biblioteca, pero no tienen acceso a la aplicación.

En cuanto a los préstamos, se podrán concretar una vez el cliente esté registrado. Se concretará la fecha máxima de devolución. Cuando se cree un libro, se generará un código QR que irá pegado a él. Este código permitirá conocer todos los datos del libro, e identificar si se encuentra en préstamo o no y las incidencias que todavía no se han resuelto.

Cabe destacar que, en caso de que sea necesario realizar una búsqueda para saber si el libro se encuentra disponible o no, BiblioCloud facilitará un sistema de búsqueda el cual devolverá los datos del libro simplemente indicando su ISBN o su título.

El trabajador también podrá llevar un control del acceso a la biblioteca. Es decir, cada vez que entre o salga cliente, indicará el sexo de dicho cliente y automáticamente el sistema guardará el acceso en la base de datos, junto con la fecha y hora en la que ha entrado o salido.

Por último, el sistema permitirá la generación de estadísticas. Esto permitirá a los trabajadores saber cuáles son los libros más prestados o la cantidad de accesos en un mes o un día en concreto.

## 1.3 Objetivos del proyecto

El objetivo principal de este proyecto es desarrollar una aplicación web que permita gestionar todo lo relacionado con la gestión de una biblioteca de forma sencilla y rápida. Este objetivo se puede desglosar en dos tipos de subobjetivos (tecnológico y empresarial), que se deben cumplir para que el proyecto tenga el éxito que se desea.

Estos son los objetivos tecnológicos:

- Permitir a los trabajadores (usuarios que tienen acceso a la aplicación web) registrar los nuevos clientes cuando vayan a realizar su primer préstamo.
- Permitir a los trabajadores llevar un control de los accesos diarios.
- Desarrollar una base de datos funcional donde se guarden todos los datos sobre los libros, los clientes, los préstamos y las incidencias, además de otras tablas secundarias pero igual de imprescindibles.
- Permitir a los administradores introducir y gestionar los datos y los permisos de los trabajadores que tienen acceso a la aplicación.
- Implementar una interfaz sencilla y completa que permita a los trabajadores realizar todas las tareas necesarias de manera más rápida. Estas tareas serían por ejemplo, el control de acceso, de préstamos o de incidencias.
- Desarrollar un sistema basado en QR que se genere una vez se cree un nuevo libro y que permita obtener todos los datos sobre este para poder identificar rápidamente su estado o incidencias.
- Generar un sistema de estadísticas y vistas que permita chequear datos generales de forma rápida y conjunta, como por ejemplo el total de accesos en un día.

Los objetivos empresariales se pueden desglosar de la siguiente forma:

- Facilitar las tareas de los trabajadores de las bibliotecas.
- Generar estadísticas que ayuden a impulsar las bibliotecas.
- Generar buenas sensaciones y confianza para optar a un próximo proyecto en común.

## 1.4 Alcance del proyecto

La aplicación informática que se va a desarrollar debe poder gestionar los libros y préstamos de estos a los clientes registrados, además de permitir la gestión de los accesos y las incidencias en las devoluciones. Para ello, el alcance de la aplicación se subdivide en funcional, estratégico e informático.

### 1.4.1 Alcance funcional

La aplicación a desarrollar deberá cubrir las funcionalidades descritas a continuación para los usuarios responsables (trabajadores y administradores):

- Permitir registrar y controlar la información de los clientes.
- Permitir una búsqueda rápida de los libros mediante el ISBN o el título para poder obtener información sobre el estado actual del mismo (disponible, no disponible, en préstamo, etc.).
- Permitir el registro de nuevos préstamos así como de nuevos clientes que todavía no hayan realizado uno.
- Permitir crear una nueva incidencia y controlar los datos de incidencias anteriores.
- Controlar la entrada de clientes a la biblioteca o centro y tener un registro de dichos accesos.
- Permitir controlar y modificar la información de todas las tablas de datos almacenadas en el sistema, además de mostrarlas a los trabajadores cuando sea necesario.
- Permitir actualizar la información sobre el estado de los libros para saber si se encuentran disponibles o no antes de realizar un préstamo.
- Permitir actualizar los datos de un préstamo cuando éste sea completado, indicando la fecha de devolución y cambiando de nuevo el estado del libro.

Cabe destacar que habrá algunas acciones que solo puedan realizar los administradores, como por ejemplo modificar la información de los trabajadores o asignar nuevos roles en caso necesario. Estas acciones estarán restringidas a los trabajadores para que no modifiquen o eliminen datos por error.

### 1.4.2 Alcance organizativo

La aplicación se compone de tres módulos principales: gestión de préstamos, control de accesos y gestión de incidencias.

- **Gestión de préstamos:** este módulo engloba los aspectos relacionados con los préstamos. Los trabajadores encargados de gestionarlos podrán listar todos los préstamos existentes o filtrarlos según el dato que deseen conocer como por ejemplo conocer qué préstamos todavía no se han devuelto o conocer cuántos préstamos ha solicitado un determinado cliente. Además de poder listar, también podrán crear nuevos préstamos y actualizarlos en caso necesario.
- **Control de accesos:** los usuarios que utilizarán este módulo principalmente serán los trabajadores. La única tarea que deberán realizar en esta sección es indicar el sexo del cliente cuando acceda a la biblioteca a la misma vez que se crea el nuevo acceso.
- **Gestión de incidencias:** este módulo también será realizado por los trabajadores y es muy parecido a la gestión de préstamos. Los trabajadores responsables podrán listar todas las incidencias que existan actualmente en el sistema, conociendo así el cliente que la ha ocasionado y una pequeña descripción explicativa. Además, también podrán crear nuevas incidencias y modificarlas.

Aunque los usuarios que principalmente utilizarán estos módulos son los propios trabajadores encargados de la gestión de la biblioteca, los administradores también tendrán acceso a ellos, con el fin de poder controlar que todo se encuentra en perfecto estado y poder detectar o corregir errores si estos surgieran.

### 1.4.3 Alcance informático

Principalmente, el sistema se conectará a una base de datos donde se almacenarán todos los datos necesarios para llevar a cabo la gestión de una biblioteca (libros, clientes, préstamos, incidencias, etc.), además de guardar también los datos sobre los usuarios que tendrán acceso y que podrán utilizar la aplicación.

Para poder lanzar la aplicación web, será necesario un servidor donde poder alojar la aplicación. Desde el propio servidor se podrá gestionar la base de datos. Tanto a la base de datos como a la gestión del servidor sólo tendrán acceso los administradores.

Será necesario disponer de acceso a distintas API para poder actualizar las tablas de la base de datos que necesiten actualización semanal o mensual, como es el caso de los libros.

También debe poder accederse o conectarse a un generador de códigos QR, que se utilizará cuando se cree un libro. Este código contendrá información sobre los datos del libro, su estado actual y sus incidencias, en caso de que tenga alguna sin resolver.

Por último, se podría implementar algún tipo de tutorial o guía sobre el uso de la aplicación, aunque en un principio no será necesario, ya que las interfaces serán bastante sencillas e intuitivas.

## **1.5 Estructura de la memoria**

El contenido de la memoria está dividido en siete capítulos, todos con importante relevancia en el proyecto.

El primer capítulo es el actual, la Introducción, donde se ha explicado la motivación y contexto del proyecto además de haber descrito y especificado sus objetivos y alcance.

El segundo capítulo contiene la planificación, en el cual se explica la metodología y tareas seguidas para llevar a cabo el proyecto además de indicar como se ha realizado la gestión de los aspectos más importantes como los costes y los riesgos.

El tercer capítulo se centra en el análisis, mostrando los diagramas creados y los requisitos necesarios para que se cumplan dichos diagramas.

El cuarto capítulo muestra el diseño, indicando cuales son las pautas seguidas para que todas las vistas de la aplicación sigan el mismo formato. También se muestran ejemplos del diseño inicial de algunas interfaces.

El quinto capítulo es la implementación, donde se explica cómo se ha ido desarrollando la aplicación y se muestran algunos resultados.

El sexto capítulo se centra en las pruebas tanto unitarias como de integración, y la validación y verificación de la aplicación así como la evaluación de las interfaces.

Y el séptimo y último capítulo trata sobre las conclusiones tanto del proyecto realizado como de los aspectos personales.

## Capítulo 2

# Planificación y seguimiento del proyecto

### 2.1 Metodología

Se decidió seguir una planificación guiada por una metodología predictiva, basada en la metodología PMBOK [4]. Se consideró que esta era la opción óptima ya que se trata de un proyecto con unas pautas que dan lugar a una estructura sencilla y secuencial, con bajo riesgo y en cascada.

Aunque desde un principio se asignaron una serie de tareas y un tiempo específico a cada una de ellas, estas han ido variando. Ha habido algunas tareas que se han podido hacer en menos tiempo mientras que otras tardaron más en realizarse debido a algunos problemas que surgieron durante su desarrollo. A pesar de los inconvenientes, siempre se ha respetado el tiempo límite establecido para el desarrollo completo del proyecto, el cual era de 300 horas.

Como ya he mencionado antes, esta metodología sigue un planteamiento en cascada tal y como se muestra en la figura 2.1. Esto permite que, en caso de que sea necesario volver a alguna de las fases anteriores para retocar algún aspecto, se pueda hacer sin ningún tipo de complicación y sin que suponga un problema para el futuro desarrollo del proyecto.

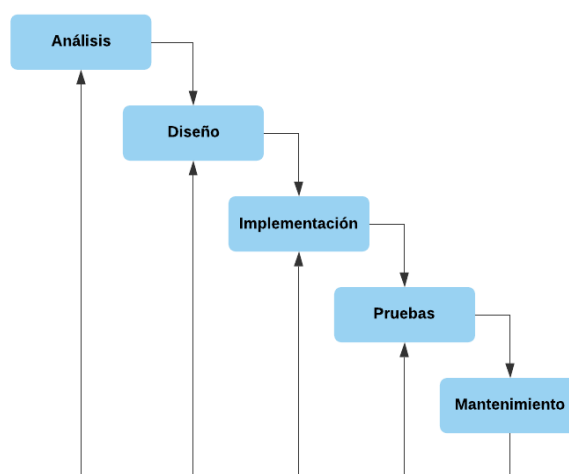


Figura 2.1 : Esquema de la metodología en cascada

A pesar de que en la figura 2.1 sólo se muestran cinco etapas principales, algunas de ellas se han dividido en varias etapas diferentes para separar mejor las tareas y tener más claros los pasos a seguir. Estas etapas y tareas están explicadas en el apartado 2.3.

## 2.2 Tecnologías y herramientas

Durante el desarrollo del proyecto se han utilizado distintas tecnologías y recursos, tanto para la programación *frontend* y *backend* de la aplicación como para la creación y gestión de la base de datos.

Como toda aplicación web, es necesario un servidor donde almacenarla. Para ello se ha hecho uso de Plesk [5][6], que permite administrar un alojamiento web y además permite acceder a la base de datos creada. Dentro de Plesk se ha registrado el nombre del dominio que utilizará la aplicación, el cual contiene la base de datos principal. Además, este servicio permite crear subdominios dependientes del dominio principal. En nuestro caso, se creará un subdominio para cada una de las bibliotecas que contraten nuestros servicios y hagan uso de BiblioCloud.

Para la gestión de la base de datos, el propio Plesk proporciona la opción de crear una nueva dentro del dominio o subdominio que queramos. En nuestro caso, la estructura de la base de datos será la misma para todos los subdominios pero cada una contendrá datos diferentes según la biblioteca a la que corresponda el subdominio. La base de datos creada se puede gestionar mediante MySQL.

En cuanto a la programación, el lenguaje principal con el que se ha desarrollado la aplicación ha sido el *Yii2 framework* [7] de PHP, acompañado de SQL para la parte del *backend*, y JavaScript, HTML, CSS y Bootstrap para dar el estilo deseado a las vistas creadas en el *frontend*. Además de estos lenguajes, también se han utilizado algunos *plugins* compatibles con *Yii2 framework*. Estos *plugins* los iré comentando a medida que surjan durante el capítulo de la implementación (Capítulo 5). Para poder instalar dichos *plugins* ha sido necesario usar tanto PuTTY [9] como FileZilla [10].



Figura 2.2 : Lenguajes de programación empleados

Los documentos y ficheros necesarios para el desarrollo de la aplicación están todos almacenados dentro de un disco duro compartido de la empresa y al cual se puede



acceder desde cualquier ordenador, por lo que no era necesario utilizar ninguna otra herramienta para poder compartir dichos ficheros, como podría ser el caso de GitHub. Para poder leer y actualizar estos ficheros, el IDE (*Integrated Development Environment*, o en español Entorno de Desarrollo Integrado) utilizado ha sido Atom.

En cuanto al equipo hardware necesario para llevar a cabo todo el desarrollo, tan solo ha sido necesario un ordenador de sobremesa con doble pantalla, ya que el trabajo ha sido desarrollado por una sola persona ayudada por otra persona supervisora.

## 2.3 Planificación temporal

Como se ha explicado anteriormente, este proyecto ha seguido una metodología predictiva donde se han definido una serie de fases y tareas. En esta sección se va a explicar cuáles son esas actividades, la estructura que siguen y el tiempo que se les ha asignado a cada una de ellas.

### 2.3.1 Fases y tareas

Las tareas del proyecto se han dividido en siete fases, basándose en la estructura de la figura 2.1 :

1. **Inicio.** Esta etapa se centra principalmente en el entendimiento del proyecto y en la definición de los objetivos y alcances que se deben cumplir.
2. **Planificación.** En esta fase se encuentran todas las tareas dedicadas a la gestión tanto del tiempo, como de los costes y recursos necesarios para desarrollar el proyecto. También se dedica tiempo a la gestión de los riesgos que puedan surgir para tener un plan de prevención en caso necesario.
3. **Análisis.** En esta sección se crean los diagramas de casos de uso y de clases, los cuales son de utilidad para tener claro en todo momento la cantidad de clases y relaciones entre ellas que debe tener BiblioCloud.
4. **Diseño.** Tal y como su propio nombre indica, en esta etapa se encuentran las actividades dedicadas al diseño tanto de las interfaces de la aplicación como de la base de datos.
5. **Implementación.** Es la fase principal del proyecto y a la que más tiempo se le dedica, ya que las actividades que se realizan en esta parte son todas aquellas relacionadas con la configuración y codificación de la aplicación.
6. **Testeo.** Antes de poner en marcha el proyecto desarrollado es necesario realizar todas las pruebas pertinentes para comprobar que no hay ningún fallo y

confirmar que se han seguido y cumplido todos los diseños previamente establecidos.

7. **Implantación.** Fase final del proyecto, donde se despliega la aplicación y se informa a los clientes de que BiblioCloud ya está listo para ser usado.

### 2.3.2 Estructura de Desglose de Trabajo

Una vez explicadas las fases, es necesario saber cuales son todas las actividades que se van a realizar en cada una de estas etapas. Para ello se ha creado un diagrama EDT (Esquema de Desglose del Trabajo), el cual se puede observar en la figura.

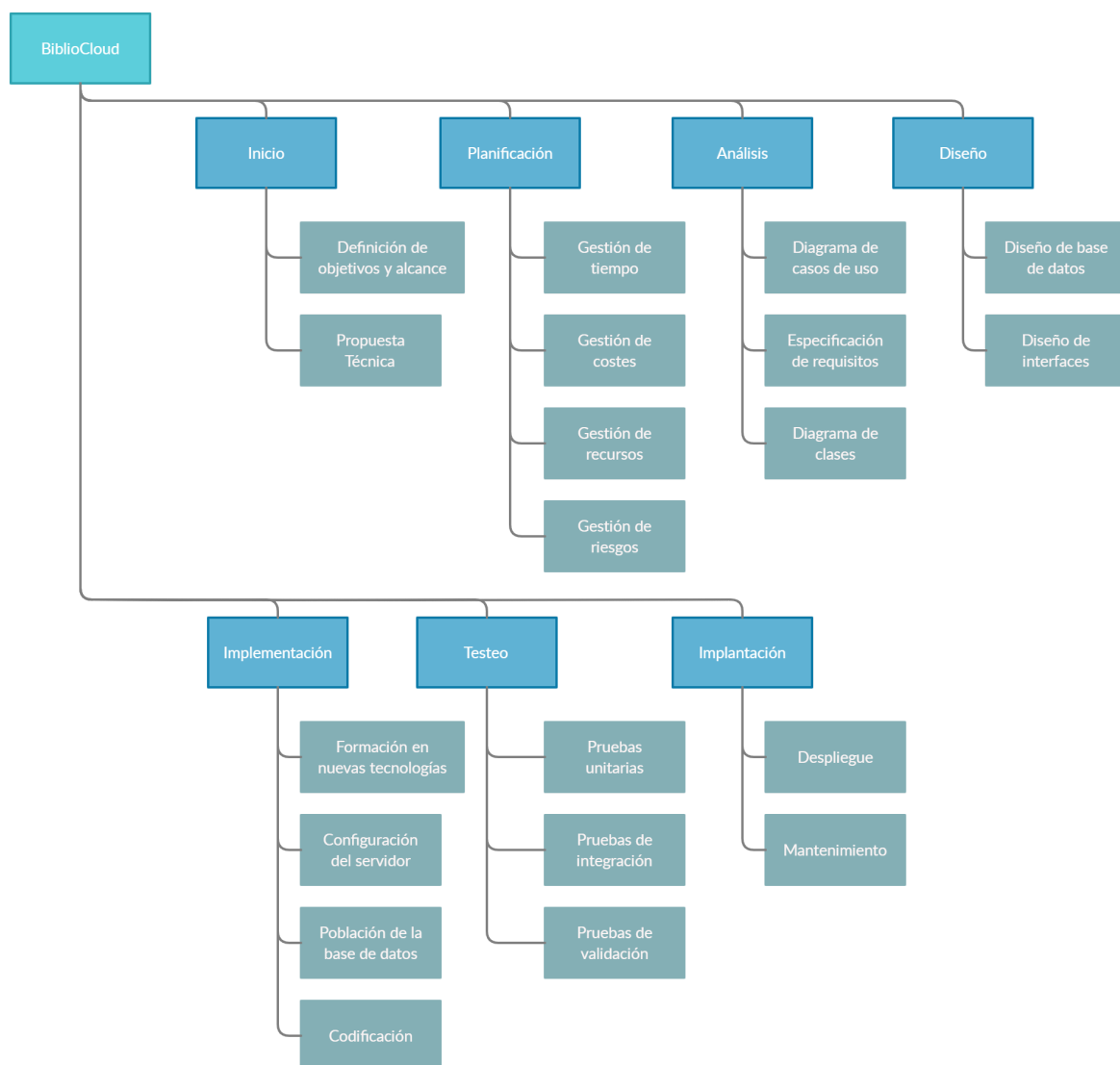


Figura 2.3 : EDT del proyecto

Cabe destacar que la tarea “Codificación” de la fase de implementación está compuesta por cinco subtareas, que no se muestran en el EDT debido a que no se podían apreciar con claridad. Estas subtareas son:

- Generación de modelos.
- Generación de vistas.
- Generación de controladores.
- Generación de interfaces.
- Implementación de funcionalidades.

### **2.3.3 Distribución temporal**

Como el proyecto se tenía que realizar en 300 horas, divididas en 25 horas semanales durante 12 semanas (60 días en total), se ha intentado ajustar el tiempo de la mejor manera posible para que el trabajo fuera completado con éxito, teniendo en cuenta la duración media habitual de cada actividad en otros proyectos similares y asignando la mayor cantidad de horas a la fase de implementación, que es la más importante.

La distribución temporal del proyecto se corresponde con la figura 2.4.

### **2.3.4 Diagrama de Gantt**

Para visualizar mejor la planificación y la dedicación temporal del proyecto, se ha creado un diagrama de Gantt, el cual se puede ver en la figura 2.5.

	i	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1			▀ Proyecto Bibliotecas	60 días	lun 01/03/21	vie 21/05/21	
2			▀ Inicio	3 días	lun 01/03/21	mié 03/03/21	
3			Definición de objetivos y alcance	1 día	lun 01/03/21	lun 01/03/21	
4			Propuesta Técnica	2 días	mar 02/03/21	mié 03/03/21	3
5			▀ Planificación	3 días	jue 04/03/21	lun 08/03/21	
6			Gestión de tiempo	1 día	jue 04/03/21	jue 04/03/21	4
7			Gestión de costes	1 día	vie 05/03/21	vie 05/03/21	6
8			Gestión de recursos	1 día	vie 05/03/21	vie 05/03/21	6
9			Gestión de riesgos	1 día	lun 08/03/21	lun 08/03/21	7;8
10			▀ Análisis	3 días	mar 09/03/21	jue 11/03/21	
11			Diagrama de casos de uso	2 días	mar 09/03/21	mié 10/03/21	9
12			Especificación de requisitos	2 días	mar 09/03/21	mié 10/03/21	9
13			Diagrama de clases	1 día	jue 11/03/21	jue 11/03/21	11;12
14			▀ Diseño	5 días	vie 12/03/21	jue 18/03/21	
15			Diseño de base de datos	3 días	vie 12/03/21	mar 16/03/21	13
16			Diseño de interfaces	2 días	mié 17/03/21	jue 18/03/21	15
17			▀ Implementación	40 días	mié 17/03/21	mar 11/05/21	
18			Formación en nuevas tecnologías	5 días	mié 17/03/21	mar 23/03/21	15
19			Configuración del servidor	1 día	mié 24/03/21	mié 24/03/21	18
20			Población de la base de datos	2 días	jue 25/03/21	vie 26/03/21	19
21			▀ Codificación	32 días	lun 29/03/21	mar 11/05/21	
22			Generación de modelos	1 día	lun 29/03/21	lun 29/03/21	20
23			Generación de vistas	3 días	mar 30/03/21	jue 01/04/21	22
24			Generación de controladores	3 días	mar 30/03/21	jue 01/04/21	22
25			Generación de interfaces	3 días	vie 02/04/21	mar 06/04/21	23;24
26			Implementación de funcionalidades	25 días	mié 07/04/21	mar 11/05/21	25
27			▀ Testeo	4 días	mié 12/05/21	lun 17/05/21	
28			Pruebas unitarias	1 día	mié 12/05/21	mié 12/05/21	26
29			Pruebas de integración	2 días	jue 13/05/21	vie 14/05/21	28
30			Pruebas de validación	1 día	lun 17/05/21	lun 17/05/21	29
31			▀ Implantación	4 días	mar 18/05/21	vie 21/05/21	
32			Despliegue	2 días	mar 18/05/21	mié 19/05/21	30
33			Mantenimiento	2 días	jue 20/05/21	vie 21/05/21	32
34			Fin del proyecto	0 días	vie 21/05/21	vie 21/05/21	33

Figura 2.4 : Distribución temporal de las tareas

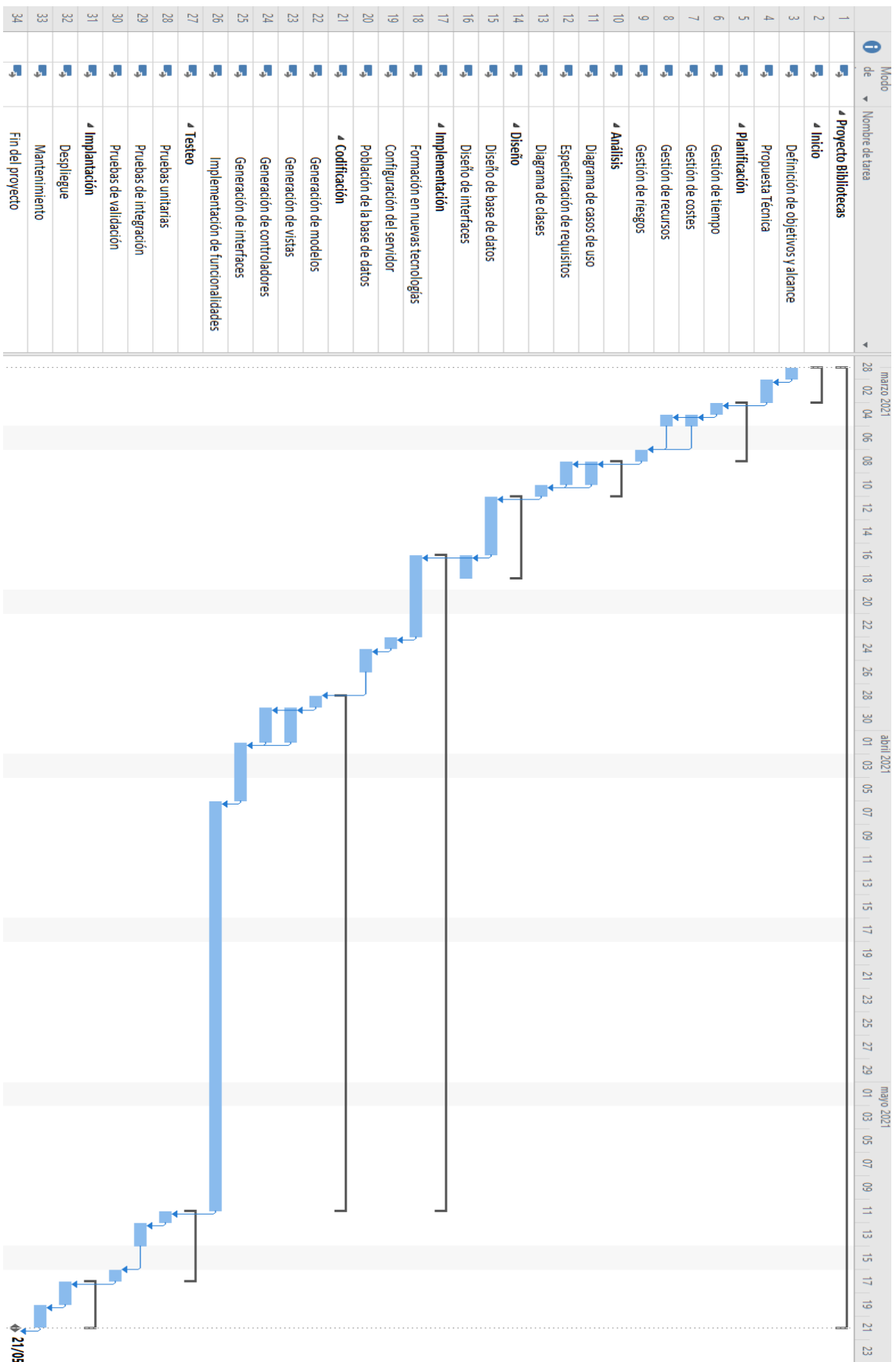


Figura 2.5 : Diagrama de Gantt

## 2.4 Estimación de recursos y costes del proyecto

Para estimar el coste de los recursos necesarios para el desarrollo de la aplicación BiblioCloud, tendremos en cuenta tres aspectos: el coste hardware, el coste software y el coste humano. Además, a estos tres tipos de costes habrá que sumarle los costes indirectos.

### 2.4.1 Costes hardware

Durante el desarrollo del proyecto han sido necesarios todos los componentes imprescindibles para un ordenador de sobremesa, los cuales son: una caja (torre) con todos los componentes necesarios con un coste total de 600€, dos monitores BenQ GW2455 valorados en 95€ cada uno, y un teclado y un ratón de 30€ y 15€ respectivamente. Todos los componentes informáticos tienen un periodo de amortización de 10 años (3650 días), excepto la caja con los componentes, que tiene un periodo medio de amortización de 5 años (1815 días).

Para calcular el coste hardware total hay que tener en cuenta el coste total amortizado de estos componentes durante el periodo de 300 horas que ha durado el proyecto. Hay que tener en cuenta que se ha trabajado 5 horas diarias, por lo que cada componente se ha utilizado un total de 60 días.

A continuación se muestra la tabla 2.1, donde se resume todo lo explicado anteriormente y se muestra el coste total.

Recurso	Precio	Amortización	Días	Coste amortizado por día	Coste total amortizado
Caja	600€	1825 días	60	0'33€ por día	19'8€
Monitor 1	95€	3650 días	60	0'026€ por día	1'56€
Monitor 2	95€	3650 días	60	0'026€ por día	1'56€
Teclado	30€	3650 días	60	0'0082€ por día	0'492€
Ratón	15€	3650 días	60	0'0041€ por día	0'246€
				<b>Total:</b>	<b>23'658€</b>

Tabla 2.1 : Costes hardware

### 2.4.2 Costes software

En cuanto a los costes software, tan solo son necesarios un servidor y el dominio. Si es verdad que el servidor tendría más sentido que estuviera en la parte de costes hardware, pero he decidido ponerlo en este apartado porque la empresa no lo controla directamente, sino que contrata sus servicios.

El servidor está almacenado en Plesk cuya tarifa es de 22€ al mes [11]. Sin embargo, nosotros hemos contratado Ionos, quien tiene un acuerdo con Plesk y el mismo tipo de servidores cuesta menos por lo que sale mucho más rentable y además evita tener que contratar otro tipo de licencias extra [12]. En nuestro caso, para este proyecto la empresa ha contratado el Servidor Cloud S, que cuesta 8€ al mes. De momento, la capacidad que proporciona este servidor es suficiente pero en caso de necesitar más capacidad se puede contratar otro más amplio sin ningún problema.

Por otro lado, el coste del dominio es de 10€ mensuales. Tanto el servidor como el dominio se han utilizado solamente para este proyecto, por tanto, el coste software total durante estos 3 meses que ha durado el proyecto es el que se muestra en la tabla 2.2.

Recurso	Coste mensual	Meses	Coste total
Servidor	8€/mes	3	24 €
Dominio	10€/mes	3	30 €
Total:			54 €

Tabla 2.2 : Costes software

### 2.4.3 Costes humanos

Para analizar los costes generados por los recursos humanos hay que tener en cuenta a todas las personas que han estado implicadas en el proyecto durante los tres meses de realización.

A pesar de que yo, siendo un alumno en prácticas no cobraba nada durante el proceso, vamos a tener en cuenta el sueldo medio de un programador junior, el cual es de 1600€ mensuales aproximadamente [13]. A este coste hay que sumarle el salario del supervisor, que es de 3400€ mensuales, ya que se trata de un desarrollador experimentado y CEO de la entidad.

El coste total de los recursos, hay que calcularlos teniendo en cuenta las horas totales dedicadas al proyecto para cada uno de los empleados. El programador junior ha trabajado las 300 horas del proyecto, mientras que el supervisor simplemente ha estado pendiente de que se cumplieran los objetivos y requisitos, por lo que ha trabajado en el proyecto tan solo 50 horas. En la tabla 2.3 queda todo mucho más claro, mostrando también el coste total de recursos humanos.

Empleado	Salario mensual	Salario por hora	Horas	Salario total
Programador junior	1600 €/mes	6'58€	300	1974€
Supervisor	3400 €/mes	13'97€	50	698'5€
			<b>Total:</b>	<b>2672'5€</b>

Tabla 2.3 : Costes humanos

A estos costes, hay que sumarle también los costes indirectos, los cuales suponen un coste de 170€ mensuales por la electricidad y 40€ mensuales por el Internet. La tabla 2.4 muestra un resumen de estos costes indirectos.

Recurso	Coste mensual	Meses	Coste total
Electricidad	170€	3	510€
Internet	40€	3	120€
		<b>Total:</b>	<b>630€</b>

Tabla 2.4 : Costes indirectos

Finalmente, en la tabla 2.5 se muestra el coste total del proyecto, el cual es de 3380'158€.



<b>Recurso</b>	<b>Coste total</b>
Costes hardware	23'658€
Costes software	54 €
Costes humanos	2672'5€
Costes indirectos	630€
<b>Total:</b>	<b>3380'158€</b>

Tabla 2.5 : Coste total del proyecto

Cabe destacar que la biblioteca también tendrá que realizar algunos gastos. Para que la aplicación funcione con toda su capacidad será necesario que la biblioteca disponga de al menos un ordenador para poder manejar la aplicación y un lector de códigos QR. Estos lectores se pueden comprar por aproximadamente 35€. Además de estos dispositivos, cada biblioteca deberá adquirir etiquetas que se puedan pegar. En estas etiquetas es donde se imprimirán los códigos QR e irán pegadas al libro que corresponda.

## 2.5 Gestión de riesgos

Este no es un proyecto en el cual puedan surgir muchas complicaciones, sin embargo no hay que confiarse ya que puede originarse algún problema en cualquier momento. Por eso, desde un principio se definieron varios posibles riesgos y la manera de prevenirlos o solucionarlos.

Uno de los principales riesgos es la poca experiencia de los empleados con las tecnologías que se van a usar para la programación. Por eso, a modo de prevención, es necesario dedicar unos días a estudiar y practicar con el nuevo lenguaje a fin de tener menos complicaciones a lo largo de la implementación.

Otro posible riesgo es la mala distribución inicial del tiempo para cada tarea. Esto se puede prevenir teniendo en cuenta los posibles problemas que puedan surgir en el momento de hacer la asignación. En caso de que el problema se haga real, la manera de poder solucionarlo es redistribuir el tiempo de cada fase o tarea. Al tratarse de una metodología en cascada, esto nos permite poder cambiar el tiempo de las fases sin que el proyecto se vea muy alterado.

Un riesgo muy común que ocurre en un porcentaje muy alto de proyectos informáticos son las complicaciones durante la implementación. En cualquier momento pueden originarse errores en la programación que no estaban previstos y que por alguna razón se tarda más en solucionar de lo previsto. Por eso es necesario establecer más tiempo a las tareas de esta fase para poder prevenir este tipo de problemas.

A parte de estos, también se podría considerar como riesgo la poca experiencia de los clientes con las tecnologías, por ello es necesario hacer unas interfaces sencillas e intuitivas.

## **2.6 Seguimiento del proyecto**

Durante el desarrollo del proyecto se ha ido comprobando que se cumplían los requisitos en todo momento y que las tareas estaban correctamente finalizadas antes de avanzar a las siguientes. El seguimiento del proyecto se puede dividir en distintas etapas, explicadas en los siguientes apartados.

### **2.6.1 Inicio del proyecto y fase de comunicación**

Al principio del proyecto, durante aproximadamente dos semanas, se dedicó tiempo a definir correctamente todos los requisitos y objetivos que debía cumplir la aplicación web BiblioCloud. Además, durante esta etapa también se diseñaron tanto la base de datos como las distintas interfaces que formarían parte de BiblioCloud.

Tras estas tareas, se dedicó una semana a la formación con el *framework* Yii2, leyendo guías, realizando tutoriales y mirando ejemplos de otros proyectos realizados por la entidad con el mismo *framework*. Durante este período fue cuando surgió el primer inconveniente. Las conexiones a Internet en el Grao de Castellón (población donde se encuentran situadas las oficinas de la empresa) fallaron un día entero, por lo que fue un día perdido. Afortunadamente, el resto de días se avanzó más rápido de lo esperado y se pudo completar la tarea en el tiempo inicialmente establecido.

### **2.6.2 Desarrollo de la implementación**

Una vez, se completaron todas las tareas previas a la implementación, se empezó a programar la aplicación siguiendo los criterios y diseños establecidos previamente. Esta etapa se fue desarrollando siguiendo un orden, es decir, primero se empezó por la gestión de los libros, seguidamente los préstamos e incidencias y finalmente los accesos y estadísticas.

Durante todo el tiempo dedicado a la implementación, que fue el más largo de todos, simplemente surgieron complicaciones en la parte de la gestión de los libros. Tuvimos un par de problemas durante la programación, lo que originó que se retrasara la tarea un poco más de lo previsto. Sin embargo, al igual que pasó con el inconveniente de la formación con Yii2, el resto de tareas de esta etapa se completaron rápidamente, por lo que pudimos recuperar el tiempo perdido y la duración del proyecto no se vio afectada. Gracias a esta rapidez en las distintas tareas, la diferencia entre el diagrama de Gantt inicial y el final es insignificante, por lo que se puede decir que el diagrama no varía.

### **2.6.3 Etapa final**

Esta etapa se llevó a cabo sin complicaciones. Se dedicó el tiempo final a la comprobación y validación de la aplicación. Aunque aparecen algunos errores durante las comprobaciones que no habían sido detectados previamente, se pudieron solucionar en el mismo instante ya que eran errores sencillos y fáciles de arreglar. Tras las pruebas pertinentes, se puso en marcha la aplicación con éxito.



## Capítulo 3

# Análisis del sistema

### 3.1 Diagrama de casos de uso

Para poder llevar a cabo el proyecto, primero es necesario identificar cuáles son los requisitos que debe cumplir la aplicación. Para ello se ha diseñado el diagrama de casos de uso de la figura 3.1, donde se muestran distintas acciones principales que deben poder ser realizadas por los usuarios que hagan uso de BiblioCloud.

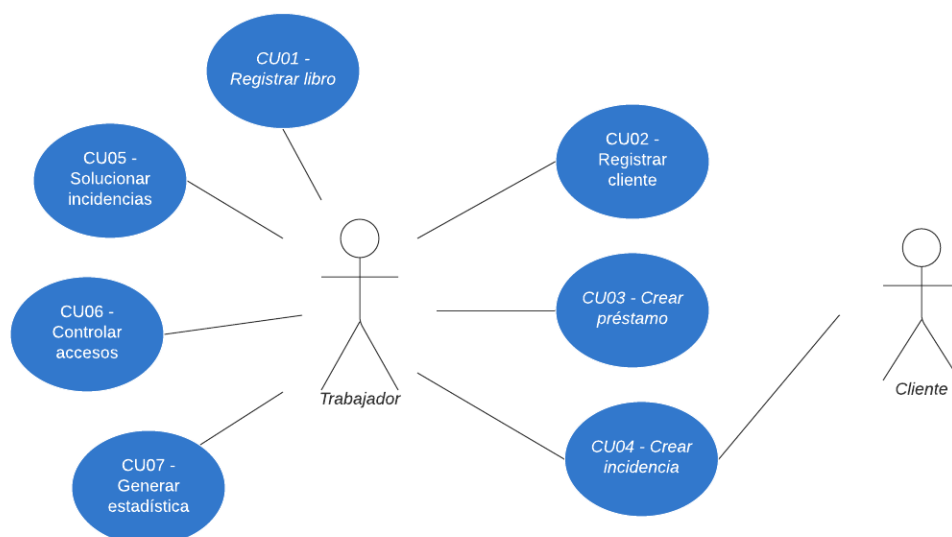


Figura 3.1 : Diagrama de casos de uso

Como se aprecia en el diagrama, existen dos actores distintos. El primero es el trabajador de la biblioteca, que es el usuario que estará registrado y que podrá utilizar todas las funcionalidades de la aplicación. Es por ello que está relacionado con todos los casos de uso. El segundo actor es el cliente que accede a la biblioteca. Aunque esté relacionado con tres casos de uso distintos, solo puede realizar directamente el CU04 (Crear una incidencia). En los otros dos casos participa indirectamente, ya que se necesitan sus datos para poder crear tanto su registro como un préstamo.

Cabe destacar que, aunque no aparezcan como casos de uso en el diagrama, también existen las funcionalidades de modificar y borrar para todos los aspectos de la biblioteca (libros, préstamos, incidencias, clientes), las cuales solo puede realizar el trabajador.

## 3.2 Requisitos funcionales

Una vez están claros cuales son todos los casos de uso, es necesario describir los requisitos funcionales de cada uno de ellos para que el proyecto cumpla con los objetivos establecidos. Estos requisitos se encuentran desglosados en las siguientes siete tablas, una por cada caso de uso.

Requisito funcional CU01	
<b>Identificador</b>	CU01
<b>Nombre</b>	Registrar libro
<b>Autor</b>	Carlos Bacete Millán
<b>Descripción</b>	El sistema debe permitir al usuario registrar un nuevo libro.
<b>Actor principal</b>	Trabajador
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Disparador</b>	Un trabajador autenticado desea registrar un nuevo libro.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la sección Libros del menú lateral.</li> <li>2. El usuario selecciona “Nuevo libro”.</li> <li>3. El usuario introduce el ISBN del libro.</li> <li>4. El sistema busca el libro mediante una API. Si lo encuentra rellena el resto de campo automáticamente. En caso contrario, el trabajador deberá hacerlo manualmente.</li> <li>5. El usuario selecciona Guardar para confirmar el registro.</li> </ol>
<b>Excepciones</b>	Si los datos del autor o la editorial no se encuentran guardados en la base de datos, el usuario deberá hacerlo manualmente antes de guardar el registro, aunque esto no suele ocurrir.
<b>Prioridad</b>	Alta

Tabla 3.1 : Requisito funcional CU01

<b>Requisito funcional CU02</b>	
<b>Identificador</b>	CU02
<b>Nombre</b>	Registrar cliente
<b>Autor</b>	Carlos Bacete Millán
<b>Descripción</b>	La aplicación debe permitir al usuario poder registrar un nuevo cliente.
<b>Actor principal</b>	Trabajador
<b>Precondición</b>	El trabajador debe estar registrado en la aplicación.
<b>Disparador</b>	Un nuevo cliente quiere hacer su primer préstamo y para ello es necesario registrarlo.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. Un nuevo cliente desea hacer un préstamo.</li> <li>2. El trabajador accede a la sección Clientes del menú lateral.</li> <li>3. El trabajador selecciona “Nuevo cliente”.</li> <li>4. El usuario introduce los datos del cliente y selecciona el botón “Guardar”.</li> <li>5. El sistema confirma el registro.</li> </ol>
<b>Excepciones</b>	El cliente ya estaba registrado previamente en la aplicación, por lo que el registro falla.
<b>Prioridad</b>	Alta
<b>Comentarios</b>	El actor cliente participa indirectamente en este caso ya que se necesitan sus datos para que pueda ser registrado.

Tabla 3.2 : Requisito funcional CU02

<b>Requisito funcional CU03</b>	
<b>Identificador</b>	CU03
<b>Nombre</b>	Crear préstamo
<b>Autor</b>	Carlos Bacete Millán
<b>Descripción</b>	El sistema debe permitir al usuario crear un préstamo sobre un libro que se encuentra disponible.
<b>Actor principal</b>	Trabajador
<b>Precondición</b>	El trabajador debe estar registrado en la aplicación.
<b>Disparador</b>	Un cliente quiere realizar el préstamo de un libro.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El trabajador accede a la sección Libros del menú lateral.</li> <li>2. El trabajador pulsa el botón de la columna “Realizar Préstamo” de la fila correspondiente al libro que se desea prestar.</li> <li>3. El trabajador introduce los datos del préstamo y presiona el botón “Guardar”.</li> <li>4. El sistema registra el préstamo.</li> </ol>
<b>Excepciones</b>	<p>El cliente no está registrado.</p> <p>El libro no está disponible debido a que se ha prestado a otro cliente.</p>
<b>Prioridad</b>	Alta
<b>Comentarios</b>	El actor cliente participa indirectamente en este caso ya que se necesitan sus datos para poder formalizar el préstamo.

Tabla 3.3 : Requisito funcional CU03



<b>Requisito funcional CU04</b>	
<b>Identificador</b>	CU04
<b>Nombre</b>	Crear incidencia
<b>Autor</b>	Carlos Bacete Millán
<b>Descripción</b>	La aplicación debe permitir tanto al trabajador como al cliente crear una incidencia sobre un libro en específico.
<b>Actor principal</b>	Trabajador, Cliente
<b>Precondición</b>	El trabajador y el cliente deben estar registrados en la aplicación.
<b>Disparador</b>	Un trabajador o un cliente quiere poner una incidencia sobre un libro.
<b>Secuencia normal</b>	<p>Si es un trabajador:</p> <ol style="list-style-type: none"> <li>1. El trabajador accede a la sección Libros del menú lateral.</li> <li>2. El trabajador pulsa el botón de la columna “Crear Incidencia” de la fila correspondiente al libro que se desea.</li> <li>3. El trabajador introduce un comentario para la incidencia y presiona el botón “Guardar”.</li> <li>4. El sistema registra la incidencia realizada por el trabajador.</li> </ol> <p>Si es un cliente:</p> <ol style="list-style-type: none"> <li>1. El cliente escanea el código QR de la etiqueta que hay pegada en el libro.</li> <li>2. El sistema le muestra una vista con los datos del libro que tiene en préstamo.</li> <li>3. El cliente pulsa el botón “Crear incidencia”.</li> <li>4. El cliente introduce un comentario para la incidencia y presiona el botón “Guardar”.</li> <li>5. El sistema registra la incidencia realizada por el cliente.</li> </ol>
<b>Prioridad</b>	Alta
<b>Comentarios</b>	El trabajador también podría hacer una secuencia similar a la del cliente si el libro estuviera en la biblioteca, aunque al escanear el código la vista que le muestra el sistema es distinta.

Tabla 3.4 : Requisito funcional CU04

<b>Requisito funcional CU05</b>	
<b>Identificador</b>	CU05
<b>Nombre</b>	Solucionar incidencias
<b>Autor</b>	Carlos Bacete Millán
<b>Descripción</b>	El sistema debe permitir al usuario poder marcar como solucionada una incidencia.
<b>Actor principal</b>	Trabajador
<b>Precondición</b>	Hay una incidencia sobre un libro sin resolver.
<b>Disparador</b>	El trabajador soluciona la incidencia y quiere hacérselo saber al sistema para que cambie su estado.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El trabajador resuelve una incidencia sobre un libro.</li> <li>2. El trabajador accede a la sección Incidencias del menú lateral.</li> <li>3. El trabajador pulsa el botón de la columna “Marcar como solucionada” de la fila correspondiente a la incidencia resuelta.</li> <li>4. El sistema cambia el estado de la incidencia a “Solucionada”.</li> </ol>
<b>Prioridad</b>	Media

Tabla 3.5 : Requisito funcional CU05

<b>Requisito funcional CU06</b>	
<b>Identificador</b>	CU06
<b>Nombre</b>	Controlar accesos
<b>Autor</b>	Carlos Bacete Millán
<b>Descripción</b>	La aplicación debe permitir registrar los accesos a la biblioteca.
<b>Actor principal</b>	Trabajador
<b>Precondición</b>	El trabajador debe estar registrado en la aplicación.
<b>Disparador</b>	Entra o sale un cliente de la biblioteca.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. Un cliente entra o sale de la biblioteca.</li> <li>2. El trabajador presiona el botón correspondiente (Entra o Sale) situado en el Control de Accesos del menú lateral.</li> <li>3. El sistema registra el acceso en caso de que sea una entrada y actualiza la fecha de salida en caso contrario.</li> </ol>
<b>Prioridad</b>	Alta
<b>Comentarios</b>	Existen un botón de entrada y salida según el sexo del cliente.

Tabla 3.6 : Requisito funcional CU06

<b>Requisito funcional CU07</b>	
<b>Identificador</b>	CU07
<b>Nombre</b>	Generar estadísticas
<b>Autor</b>	Carlos Bacete Millán
<b>Descripción</b>	El sistema debe permitir al usuario generar estadísticas a partir de los datos guardados en la base de datos.
<b>Actor principal</b>	Trabajador
<b>Precondición</b>	El trabajador debe estar registrado en la aplicación.
<b>Disparador</b>	El trabajador quiere generar estadísticas.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El trabajador accede a la sección Estadísticas del menú lateral.</li> <li>2. El trabajador selecciona el tipo de estadísticas que desee (estadísticas de acceso o estadísticas de préstamos).</li> <li>3. El trabajador rellena los campos necesarios y presiona el botón de generar estadísticas.</li> <li>4. El sistema utiliza los datos de los campos para buscar en la base de datos los campos que cumplan las características deseadas y muestra los resultados al trabajador.</li> <li>5. El trabajador visualiza las estadísticas y las imprime.</li> </ol>
<b>Prioridad</b>	Alta
<b>Comentarios</b>	No siempre es necesario imprimir las estadísticas.

Tabla 3.7 : Requisito funcional CU07

### 3.3 Diagrama de clases

A continuación se muestra el diagrama de clases creado a partir de las distintas necesidades y los diversos requisitos del sistema. Además, este diagrama servirá de base para la creación y diseño de la base de datos de la que hará uso la aplicación BiblioCloud.

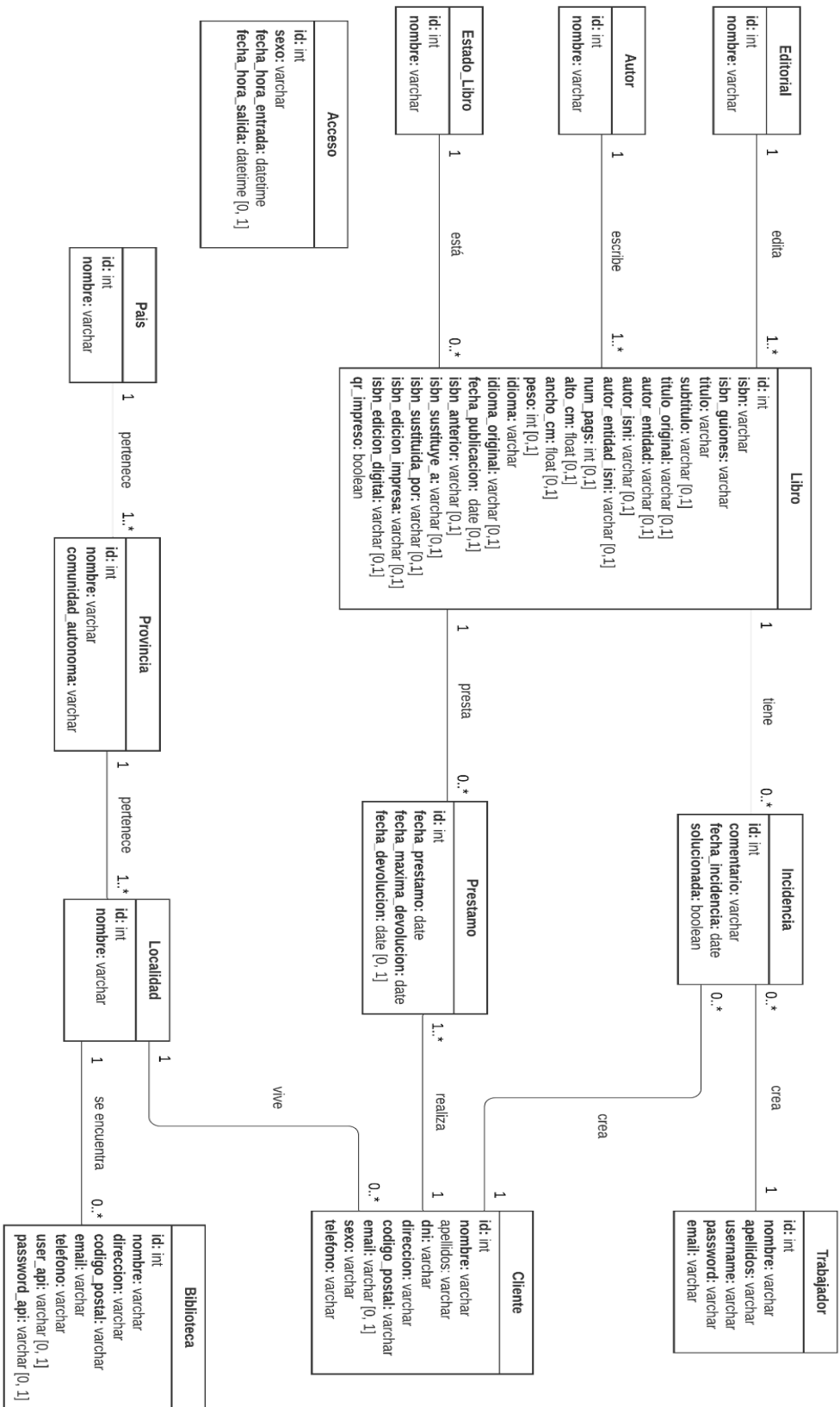


Fig 3.2 : Diagrama de clases

Como se puede observar en la figura 3.2, hay una clase (Acceso) que no se relaciona con ninguna otra. Aunque pueda parecer que está mal y que falta relacionarla con otra clase como por ejemplo clientes, no es así. Los accesos van por separado al resto de clases, ya que solo sirven para poder generar estadísticas y no es necesario que se relacionen con las demás clases.

En cuanto a los atributos de las clases, hay que hacer una pequeña mención al atributo nombre de la clase Estado\_Libro. Este atributo solo puede tomar los valores de Disponible o En préstamo.

Por último, cabe remarcar que aunque parezca necesario tener una relación entre las clases Trabajador y Biblioteca, se puede prescindir de esta ya que cada biblioteca estará en un subdominio distinto, por tanto ya se sabe que los trabajadores pertenecen a esa biblioteca.

## Capítulo 4

# Diseño del sistema

### 4.1 Diseño de la arquitectura del sistema

La arquitectura del proyecto sigue la estructura MVC (Modelo - Vista - Controlador) [14], un patrón de diseño muy utilizado en los proyectos informáticos que permite separar los datos de la aplicación, la interfaz de usuario, y la lógica de control.

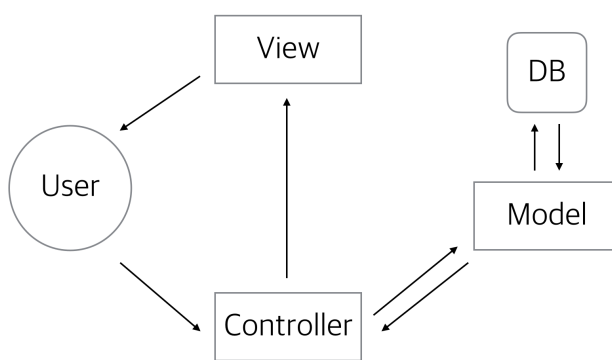


Figura 4.1 : Patrón de diseño MVC. Fuente: Seul Lee [15]

Como se puede apreciar en la figura 4.1, el patrón de diseño MVC está separado en tres capas distintas, que son:

- **Modelo:** Se encarga de gestionar la información y los datos del sistema, consultando a la base de datos.
- **Vista:** Es la representación visual de los datos, por tanto, se encarga de dar forma a la interfaz que verá el usuario.
- **Controlador:** Se encarga de la comunicación entre la vista y el modelo. Recibe peticiones del usuario desde la vista, envía y solicita los datos al modelo, y devuelve el resultado a la vista.

En cuanto a la arquitectura del servidor, tal y como se ha comentado en el apartado 2.2 de la memoria, habrá un dominio principal que contendrá la base de datos inicial. Dentro de este subdominio, se crearán distintos subdominios, uno para cada una de las bibliotecas que contraten los servicios de Inttegrum y hagan uso de BiblioCloud. Cada

subdominio tendrá la misma base de datos que el dominio principal, pero cada una de ellas contendrá datos distintos relativos a la biblioteca a la que corresponda.

## **4.2 Diseño de la base de datos**

El diseño de la base de datos surge a partir del diagrama de clases creado previamente, siendo cada clase una tabla de la base de datos con los mismos atributos sumándole a estos las claves ajenas de las relaciones, siempre y cuando sean necesarias.

Además de una tabla para cada clase, también es necesario crear una tabla auxiliar llamada “libros\_espanya”. Esta tabla contendrá todos los libros que hay registrados en España. Además, cuando un trabajador quiera registrar un nuevo libro, introducirá el ISBN y el sistema consultará esta tabla para rellenar el resto de campos automáticamente. Es por eso que esta es una tabla auxiliar.

### **4.2.1 Población previa de la base de datos**

Una vez diseñada y creada la base de datos que va a utilizar el sistema, es necesario poblar algunas tablas con datos antes de empezar con la implementación. Una de estas tablas, como se ha comentado previamente, es la tabla auxiliar “libros\_espanya”. Esta tabla contiene todos los libros registrados en la Agencia del ISBN [16]. Esta agencia dispone de una página web llamada Dilve [17], que es donde se encuentra su base de datos con todos los libros que tienen registrados. Dilve permite descargar y hacer uso del contenido de su base de datos [18], pero para ello hay que avisarles previamente para que puedan comprobar cuales son las intenciones que tiene el cliente.

Las otras tablas en las que hay que introducir datos previamente, son las tres tablas referidas a sitios geográficos (país, provincia y localidad). Para ello, se ha accedido a la página web de la Agencia Tributaria Española [19], que contiene datos de todos los municipios de España y, al ser de uso público, permite descargar toda la información necesaria sin ningún tipo de problema.

## **4.3 Diseño de las interfaces de usuario**

Aunque no se diseñó ningún prototipo como tal, sí que se pensaron el tipo de interfaces que se quería hacer y los estilos que debían seguir cada una de ellas, de manera que resultara cómodo para la vista del usuario y fuera intuitivo y fácil de usar.

En la aplicación se pueden destacar tres tipos de interfaces distintas: los índices de cada sección, los formularios de creación y modificación, y la vista para las estadísticas.



Para las interfaces de los índices de cada sección, se decidió que estuvieran formadas por una tabla (*grid*) que contuviera los datos más importantes de cada sección, similar a la figura 4.2.

Nombre	Localidad	Direccion	Codigo Postal	Email	Telefono	Acciones
Biblioteca	CASTELLÓ DE LA PLANA	Calle de Integrum	12000	email@email.email	987654321	

Figura 4.2 : Vista índice con *grid*

El formato de estos es igual para todas las secciones, un color azulado para el *grid* con una columna por cada dato más una columna final de acciones con botones para cada acción que se pueda usar. Estos botones son iconos obtenidos de Font Awesome [20]. Los iconos utilizados en todos los *grid* para las acciones son los de la figura 4.3, aunque también se han utilizado otros distintos pero tan solo aparecen una o dos veces en toda la aplicación. Todos los iconos tienen un color y tamaño igual a los de la figura 4.3.



Fig 4.3 : Iconos de la columna de acciones

En cuanto a los formularios, todos los campos tienen la misma anchura y cada formulario está diseñado para que quepa entero en la pantalla. De este modo evitamos que el usuario tenga que desplazarse arriba o abajo y pueda ver todos los datos que ha introducido de un solo vistazo. Además al final de cada formulario hay un botón para guardar toda la información introducida. Todos los botones de la BiblioCloud siguen el sistema de botones proporcionado por Bootstrap [21] (figura 4.4), donde cada botón adquiere un color dependiendo de su finalidad.

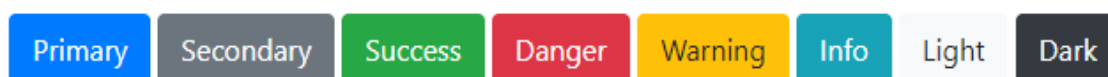


Fig 4.4 : Botones Bootstrap

Para el último tipo de interfaces, el de las estadísticas, se pensaron dos diseños distintos:

- Si se trata de un ranking, el sistema muestra una vista formada por una tabla HTML con los datos del ranking.
- En caso contrario, el sistema muestra una vista con dos gráficas (una de líneas y otra circular) en las cuales se diferencian los datos entre hombres y mujeres, ya que a cada uno se le asigna un color diferente.

Aparte de los de las interfaces, existe otro diseño muy importante en BiblioCloud ya que es visible en todas las vistas de la aplicación: el del menú lateral y la barra superior. La barra superior superior muestra el logo de la aplicación en la parte izquierda y el nombre del usuario en la parte derecha, a partir del cual se puede acceder al perfil de usuario. Esta barra es de color azul color que predomina en toda la aplicación, ya que, por ejemplo, todos los *grids* son del mismo color.

El color del menú lateral es grisáceo oscuro pero sin llegar a ser negro. Este menú está dividido en tres secciones: un control de accesos con botones HTML de color gris clarito, una barra de búsqueda similar a la de los formularios, y un menú principal donde hay una opción para cada sección junto a un icono que la representa.

## Capítulo 5

# Implementación

### 5.1 Detalles de la implementación

El *framework* Yii2 permite crear los archivos necesarios para cada una de las capas del patrón de diseño MVC. Previamente, se habrá indicado cuál es la base de datos con la que se relaciona la aplicación. A partir de esa base de datos, para cada una de las tablas que la componen se puede crear el modelo y, una vez esto está insertado, se crea el controlador con las funciones CRUD (Create, Read, Update, Delete) y sus vistas relacionadas. En la figura 5.1 se puede apreciar los paquetes para los controladores y las vistas con los ficheros básicos, y en la figura 5.2, se muestran los paquetes para los modelos y sus respectivos archivos de búsqueda. Todos estos paquetes han sido generados automáticamente por Yii2 para la aplicación BiblioCloud.

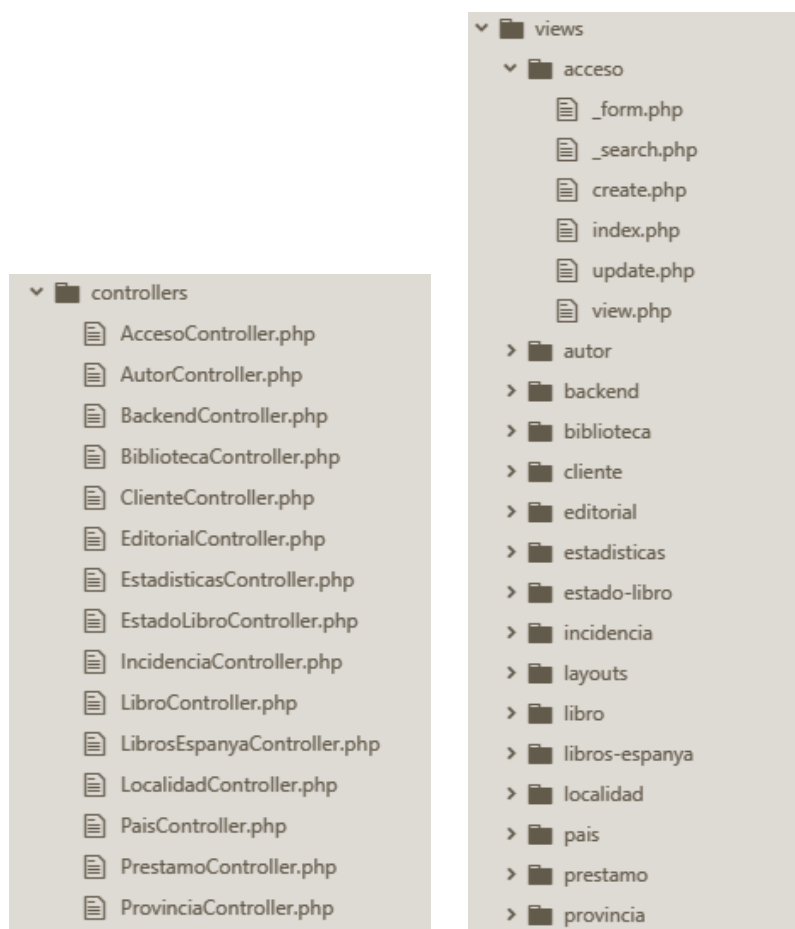


Figura 5.1 : Paquetes de controladores y vistas

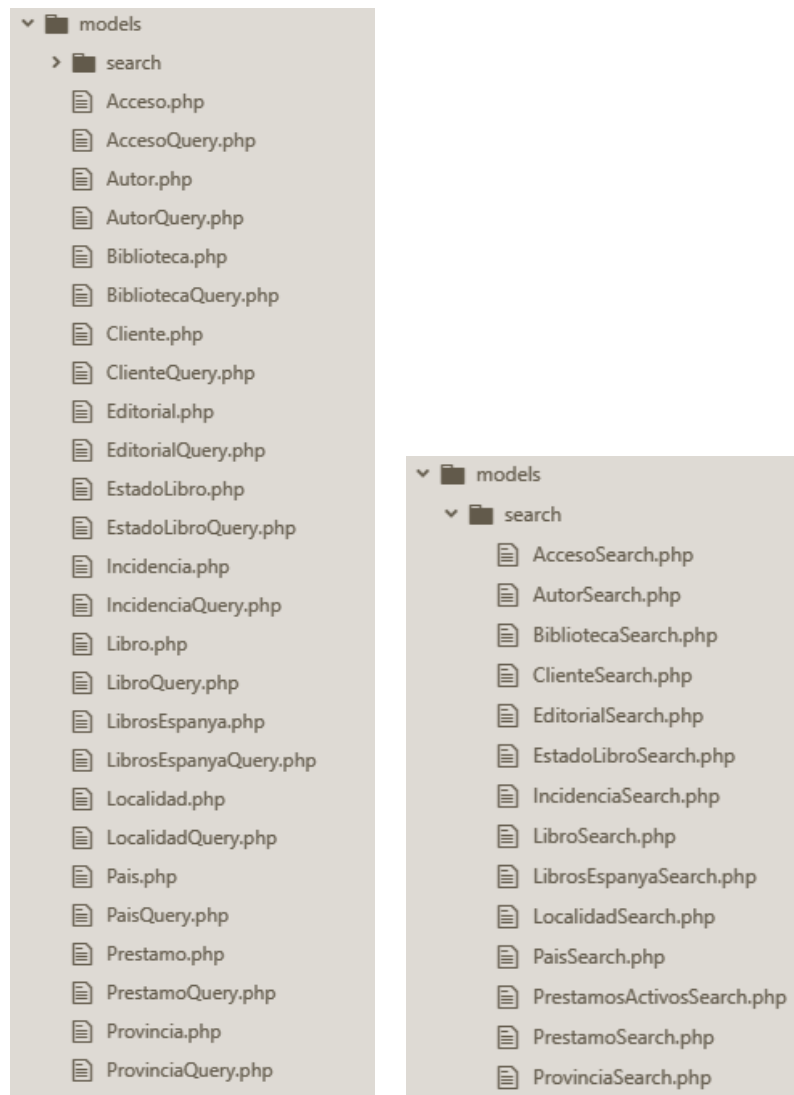


Figura 5.2 : Paquetes de modelos y búsqueda

Yii2 solo genera la estructura básica de cada uno de estos ficheros, por lo que hay que retocarlos para que se adecuen a las necesidades del proyecto. Por ejemplo, las vistas hay que modificarlas para que sigan el formato establecido en los prototipos y en los controladores hay que añadir nuevas funciones para poder implementar las funcionalidades que debe cumplir la aplicación. Todos estos cambios se cuentan al detalle en el siguiente apartado.

## 5.2 Desarrollo y resultados

Como ya he comentado anteriormente, Yii2 genera automáticamente los ficheros necesarios para el patrón de diseño MVC. Sin embargo, estos archivos son muy básicos, por lo que es necesario retocarlos para adecuarlos a los requisitos establecidos.

Por ello, hay dos archivos que se han modificado de la misma manera para todos los modelos. El primero de ellos, es el formulario creado por el fichero `_form` (ver la figura 5.1) que se lanza cuando se quiere crear o modificar alguno de los modelos. A esta vista se le ha intentado dar el mismo formato para cada modelo, realizando algunas modificaciones dependiendo de la longitud necesaria para los campos del formulario. En la figura 5.3 se muestra como es el formato definitivo de un formulario, cogiendo como ejemplo el que se muestra cuando se quiere crear un cliente.

Figura 5.3 : Formulario

El otro archivo que se ha modificado para seguir la misma estructura en todos los modelos es el fichero `index.php` (ver la figura 5.1). Este archivo muestra un *grid* con los datos del modelo y algunos botones para lanzar diferentes funcionalidades. La figura 5.4 muestra un ejemplo del *grid* generado para el index de biblioteca, que se ha implementado utilizando el *plugin* DynaGrid para Yii2 proporcionado por Krajee [22], este *plugin* permite darle un formato a la tabla con mucha facilidad y además proporciona al usuario distintas funcionalidades como modificar las columnas del *grid*, reordenarlas o incluso descargar todos los datos que contenga.

Además, cada *grid* contiene una última columna con varias acciones que se pueden realizar con cada fila de la tabla. Como se puede apreciar en la columna Acciones del *grid* de la figura 4.2, existen 4 botones, cada uno de ellos asociado a distintas acciones sobre el modelo al que pertenece la fila. Estos botones, como se ha explicado en la parte de diseño de interfaces, son los de la figura 4.3. De izquierda a derecha, los botones sirven para generar un PDF con todos los datos de la fila (similar al que se muestra en la figura 5.4, relacionado con una incidencia), ver los datos de la fila, modificar los datos (botón que lanza al formulario explicado previamente) y eliminar la fila. Cabe destacar que tanto los botones de esta columna como los del resto de la aplicación son iconos gratuitos proporcionados por Font Awesome [20].

## Datos de la incidencia

Libro	A algunos les gustan frías
Cliente	(no definido)
Comentario	Incidencia 1
Fecha	2021-05-17
Trabajador	administrador
Solucionada	Sí

Figura 5.4 : PDF generado con los datos del modelo seleccionado

En los siguientes apartados se comenta cómo se han ido desarrollando cada uno de los aspectos de la aplicación BiblioCloud. A pesar de que los apartados se muestran separados por secciones para que todo tenga sentido y quede mucho más clara la explicación, algunos apartados se han ido desarrollando simultáneamente ya que sus funcionalidades estaban relacionadas. Por ejemplo, la gestión de préstamos e incidencias se ha realizado a la vez que la gestión de libros.

### 5.2.1 Menú principal

El menú principal es el que se puede ver en la figura 5.5. Como se puede apreciar, en la parte central contiene estadísticas básicas sobre diversas acciones de la biblioteca. Todas estas estadísticas se han podido sacar realizando consultas a la base de datos desde el controlador y enviando los resultados a la vista para que los muestre al cliente. Los rectángulos de colores de la parte superior indican el total de libros registrados, los libros disponibles, los libros en préstamo y las incidencias sin resolver, de izquierda a derecha. Además, pulsando sobre estos rectángulos, se redirige al usuario a una vista que muestra un *grid* con los datos sobre la estadística. Por ejemplo, si hubiera dos incidencias sin resolver, se mostraría un *grid* con los datos de las dos incidencias cuando se pulse sobre la estadística.

Además, también en la parte central se muestran un ranking con los libros más prestados por la biblioteca y un *grid* con los accesos del día actual.

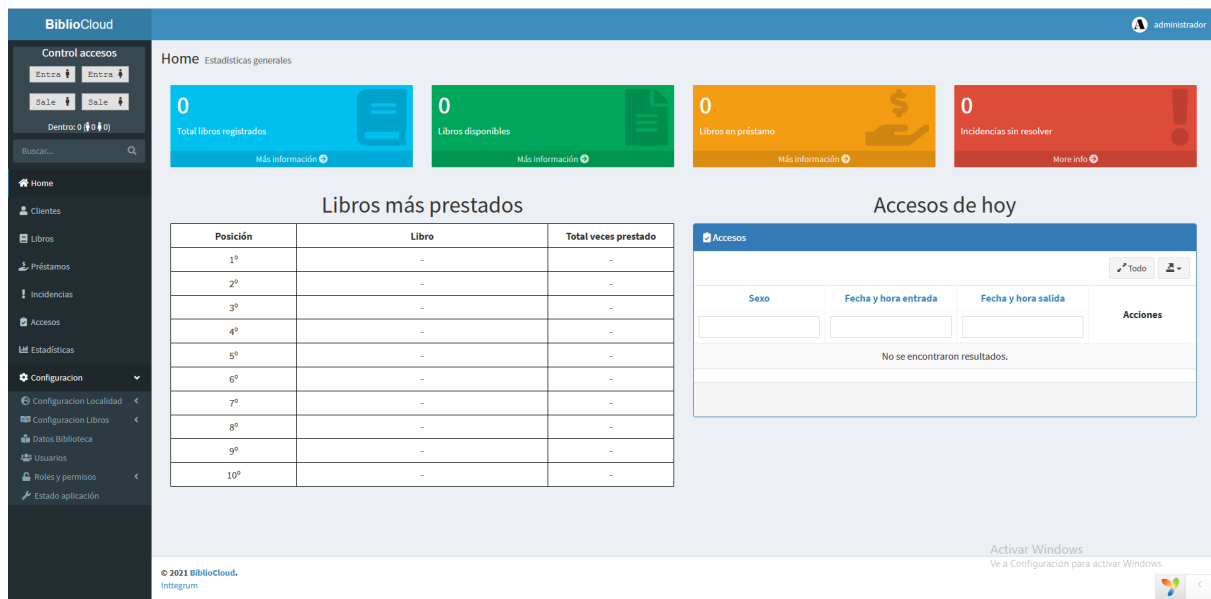


Figura 5.5 : Menú principal

El menú principal dispone de dos barras con distintas opciones: una barra superior y una barra lateral. Estas dos barras aparecen en todas las vistas de la aplicación, por lo que son accesibles en cualquier momento.

En la superior simplemente se muestra el nombre del usuario y su avatar. Si se pulsa, aparece un menú con opciones para ir al perfil o salir. En el menú lateral (figura 5.6) se muestra el nombre de la aplicación arriba del todo, y luego está dividido en tres secciones: control de accesos, barra de búsqueda y menú.

El control de accesos lo explico en el apartado 5.2.5 (Gestión de accesos). La barra de búsqueda sirve para buscar algún libro en específico que contenga el valor introducido en el campo en cuatro de sus atributos: ISBN, título, autor o editorial. Cuando se realiza la búsqueda, se redirige al usuario a una vista que muestra los resultados encontrados para cada uno de los atributos, separando los resultados en cuatro pestañas (*tabs*) distintos tal y como aparece en la figura 5.7, los cuales son generados utilizando el plugin Tabs-X proporcionado por Krajee [23]. Esta separación por *tabs* permite al usuario identificar mejor en qué atributo del libro se ha encontrado la coincidencia con el valor buscado. En el ejemplo mostrado en la figura 5.7 se muestran los resultados de los libros encontrados que contienen la palabra “acantilado” en su editorial.





## 5.2.2 Gestión de libros

La gestión de los libros es la parte principal del sistema, ya que se relaciona con prácticamente todo. Para empezar, cuando se selecciona la opción libros del menú de la barra lateral, se lanza la vista con el *grid* que muestra los datos de todos los libros que dispone la biblioteca. Aunque como ya he comentado, los *grid* son todos similares, en el de los libros hemos añadido varias modificaciones para disponer de más funcionalidades necesarias para su gestión.



The screenshot shows a web interface for managing books. At the top, there's a header with 'Libros' and a breadcrumb 'Home > Libros'. Below the header, there are two buttons: '+ Nuevo' (green) and 'Imprimir varios QR' (blue). A message states: 'Si deseas crear un préstamo o una incidencia, busca el libro que desees y pulsa el botón de la columna correspondiente'. Below this is a table titled 'Libros' with a sub-header 'Mostrando 1-2 de 2 elementos.' and a button 'Imprimir Seleccionados'. The table has columns: 'Seleccionar', 'Isbn', 'Titulo', 'Editorial', 'Autor', 'Estado', 'Qr impreso', 'Qr', 'Realizar Préstamo', 'Crear Incidencia', and 'Otras Acciones'. There are two rows of data, both for the book 'A algunos les gustan frías' by Lardner, Ring, published by Acantilado. The first row is 'Disponible' and the second is 'En préstamo'. Both rows have a green checkmark in the 'Qr impreso' column and a QR code icon in the 'Qr' column. The 'Realizar Préstamo' and 'Crear Incidencia' columns have icons for these actions. The 'Otras Acciones' column has icons for viewing, editing, and deleting.

Seleccionar	Isbn	Titulo	Editorial	Autor	Estado	Qr impreso	Qr	Realizar Préstamo	Crear Incidencia	Otras Acciones
<input type="checkbox"/>	9788495359360	A algunos les gustan frías	Acantilado	Lardner, Ring	Disponible	✓	QR	👤	!	🔍 📄 🗑️
<input type="checkbox"/>	9788495359360	A algunos les gustan frías	Acantilado	Lardner, Ring	En préstamo	✓	QR	👤	!	🔍 📄 🗑️

Figura 5.8 : *Grid* de libros

A parte de las columnas de los datos principales de cada libro y la columna con las acciones básicas explicadas en el apartado 5.2, este *grid* dispone de tres columnas de acciones más con sus botones correspondientes para poder realizarlas. Dos de ellas sirven para crear un préstamo o una incidencia sobre un libro, y la otra acción sirve para ver el código QR del libro, mostrado en un modal que además proporciona la opción para imprimir dicho código. Para poder generar se ha hecho uso del código abierto y de libre uso proporcionado en el GitHub de 2 amigos [24]. Los QR se imprimen en etiquetas y se pegan en los libros. Cada uno de estos códigos contiene un enlace que abre la vista con los datos del libro. Esta vista es la de la figura 5.12 o 5.13, dependiendo de cual sea el estado actual del libro (Disponible o En préstamo).

Como se puede apreciar en la figura 5.8, dentro del propio *grid* y antes de mostrar los datos de cada libro, hay un botón que sirve para imprimir el QR de todas las filas seleccionadas (para seleccionarlas se usa la primera columna del *grid*). Este botón sólo funcionará cuando se seleccionen entre 1 y 24 filas, ya que este es el límite de etiquetas en una hoja [25]. Las etiquetas se pueden adquirir en la página web de Herma [26]. Cuando se pulsa el botón se genera un PDF con todos los QR a imprimir, siguiendo el formato de la plantilla de las etiquetas [25]. Para crear el PDF, se utiliza el *plugin* MPdf [27].

Fuera del *grid* aparecen dos botones. Uno de ellos sirve para imprimir el código QR de los primeros 24 libros en los que todavía no se ha impreso el código. El otro botón sirve para crear un nuevo libro y redirige a la vista con el formulario de la figura 5.9.

Crear Nuevo Libro

ISBN \*  
Introduce el isbn sin guiones (13 caracteres)

Por favor, introduce primero el ISBN del nuevo libro. El resto de campos se rellenarán automáticamente. En caso de que esto no ocurra, tendrá que hacerlo manualmente.

Título \*  
Subtítulo

Editorial \*  
Seleccione Uno

Si la editorial no se encuentra en la lista, creala utilizando el siguiente botón. Una vez creada, vuelvela a introducir en el campo del formulario.

+ Nueva editorial

Autor \*  
Seleccione Uno

Si el autor no se encuentra en la lista, crealo utilizando el siguiente botón. Una vez creado, vuelvelo a introducir en el campo del formulario.

+ Nuevo autor

Idioma \*  
Fecha publicación

Estado libro \*  
Disponibile

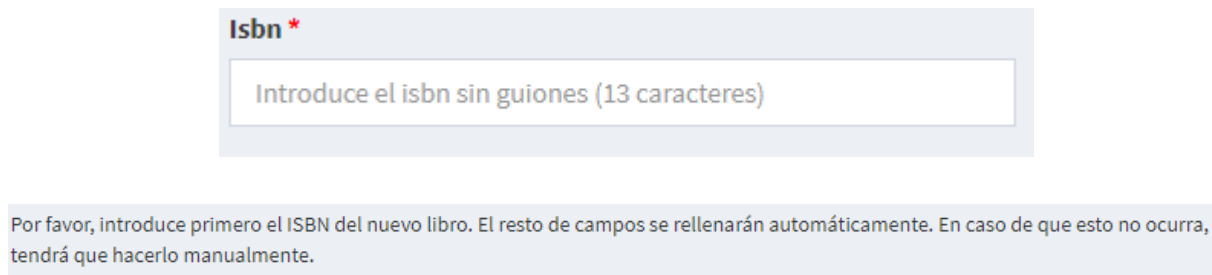
Número de páginas

Save

Figura 5.9 : Formulario para dar de alta un libro

Este formulario contiene varios campos creados mediante *plugins* disponibles para Yii2 proporcionados por Krajee. En concreto, se han utilizado dos *plugins* distintos. El primero es el *plugin* Select 2 [28], utilizado para los campos editorial y autor, que muestra todos los autores o editoriales existentes en la base de datos. El otro es el *plugin* DateControl [29], que permite seleccionar una fecha escogiendo en un calendario.

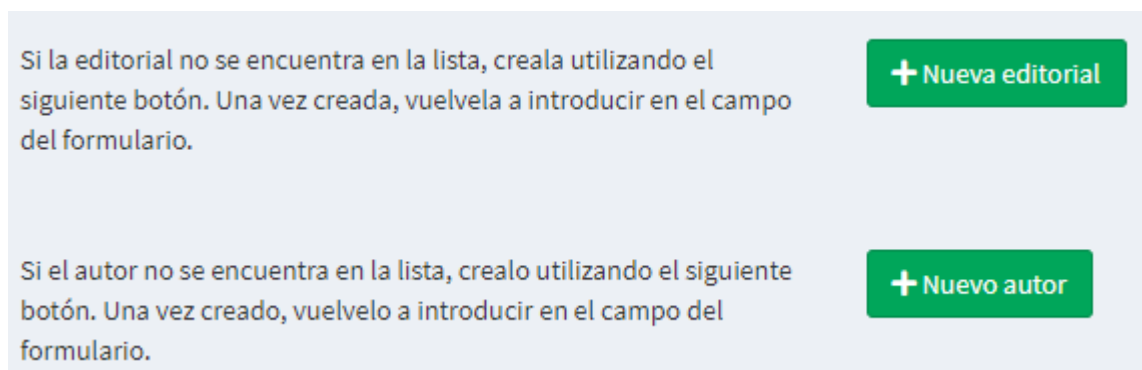
Como se puede ver en la figura 5.9 y en la figura 5.10, todos los campos están bloqueados excepto el del ISBN, que es el primero. Esto está hecho así para facilitar la tarea al usuario, que solamente debe introducir el ISBN del nuevo libro y el resto de campos se rellenan automáticamente controlando los elementos HTML mediante JavaScript. Para que esto sea posible, se usan dos métodos distintos. El primero de ellos es consultando la tabla “libros\_espanya” de la base de datos, que contiene los datos de todos los libros que había disponibles en la base de datos de Dilve [16], los cuales se descargaron e insertaron en dicha tabla antes de empezar con la implementación de la aplicación. En caso de que el ISBN no se encuentre en esta tabla, se pasa a la segunda opción, consultar la base de datos de Dilve mediante la API que proporciona [30]. Si en cualquiera de los dos casos no se encuentra el libro correspondiente al libro utilizado, el usuario deberá introducirlo manualmente.



The image shows a light blue rectangular box representing a form field. At the top left, the text "Isbn \*" is displayed in a dark font. Below this, there is a white rectangular input area with a thin border. Inside the input area, the placeholder text "Introduce el isbn sin guiones (13 caracteres)" is written in a light gray font. Below the input area, there is a light blue horizontal bar containing the following text: "Por favor, introduce primero el ISBN del nuevo libro. El resto de campos se rellenarán automáticamente. En caso de que esto no ocurra, tendrá que hacerlo manualmente."

Figura 5.10 : Campo ISBN del formulario de libros

Por último, aparte del botón para guardar los datos introducidos en el formulario, existen dos botones para poder crear al autor o a la editorial en caso de que no existan previamente y no se puedan introducir mediante el *plugin* Select2. Cada uno de estos botones despliega un modal que contiene el formulario para poder crear el nuevo autor o la nueva editorial.



The image shows a light blue rectangular box containing two rows of text and buttons. The first row has the text "Si la editorial no se encuentra en la lista, creala utilizando el siguiente botón. Una vez creada, vuelvela a introducir en el campo del formulario." followed by a green button with a white plus sign and the text "Nueva editorial". The second row has the text "Si el autor no se encuentra en la lista, crealo utilizando el siguiente botón. Una vez creado, vuelvelo a introducir en el campo del formulario." followed by a green button with a white plus sign and the text "Nuevo autor".

Figura 5.11 : Botones que despliegan modals

### 5.2.3 Gestión de préstamos e incidencias

Tanto para hacer un préstamo como para hacer una incidencia, el libro debe estar creado previamente. Al igual que en la gestión del libro, cuando se pulsa sobre su opción sobre el menú de la barra lateral, se redirige a una vista con el *grid* correspondiente. En este caso, el de incidencias contiene una nueva columna con un botón que sirve para marcar como solucionada una incidencia.

Los formularios son similares a los ya comentados previamente. El único cambio destacable ocurre en el formulario del préstamo. En este, hay que introducir el dni del cliente, pero, en caso de que dicho cliente todavía no esté registrado en la aplicación, se proporciona un botón que redirige al formulario del cliente para poder efectuar dicha acción.

Una vez se crean las incidencias y préstamos de un libro, se puede consultar la vista de este para saber sus datos. Esta vista se puede ver, pulsando el botón correspondiente en la columna de acciones del *grid* de Libros o escaneando el código QR del libro que se quiera consultar. En el caso de que el libro se encuentre disponible (figura 5.12), se muestran datos generales sobre el libro y el total de préstamos e incidencias que se han hecho sobre él separados por *tabs* [23] que contiene un *grid* con los datos que correspondan. Cabe destacar que primero se muestran las incidencias que no están solucionadas. Además, se proporcionan dos botones: uno para realizar un préstamo y crear una incidencia.

En el caso de que el libro se encuentre en préstamo (figura 5.13), la vista cambia ligeramente. Esta vez, además de los datos generales del libro, se muestra la información del préstamo actual. Además, también se muestran dos botones: uno para crear una incidencia y el otro para marcar el libro como devuelto.

Estado del libro: Disponible

Realizar préstamo | Crear incidencia

Incidencias totales (3) | Préstamos anteriores (0)

Mostrando 1-3 de 3 elementos.

Comentario	Libro	Fecha	Cliente	Trabajador	Solucionada	Marcar como solucionada	Acciones
Incidencia 2	9788495359360	2021-05-17	(no definido)	administrador	✗	👍	🔍 🔄 🗑️
Incidencia 1	9788495359360	2021-05-17	(no definido)	administrador	✓	👍	🔍 🔄 🗑️
Incidencia 3	9788495359360	2021-05-17	(no definido)	administrador	✓	👍	🔍 🔄 🗑️

Datos generales del libro

Isbn	9788495359360
Título	A algunos les gustan frías
Subtítulo	
Editorial	Acantilado
Autor	Lardner, Ring
Numero de paginas	256
Idioma	spa

Activar Windows  
Ve a Configuración para activar Windows.

Figura 5.12 : Vista del libro disponible

Estado del libro: En préstamo

Devolver libro

Crear incidencia

Datos del préstamo

Prestado a	Prueba, Cliente
Fecha del préstamo	2021-05-17
Fecha máxima de devolución	2021-05-29

Datos generales del libro

Isbn	9788495359360
Título	A algunos les gustan frías
Subtítulo	
Editorial	Acantilado
Autor	Lardner, Ring
Numero de paginas	256
Idioma	spa
Fecha publicacion	2001-01-01

Otros datos

Isbn con guiones	978-84-95359-36-0
Autor entidad	
Altura	21
Anchura	13
Peso	320

Figura 5.13 : Vista del libro en préstamo

## 5.2.4 Gestión de clientes

La gestión de clientes es similar a la de incidencias o préstamos. Al igual que ocurría en el caso anterior, cuando se pulsa el botón correspondiente en el menú lateral, se redirige a una vista que muestra un *grid* con todos los clientes. Además, el formulario para crear clientes también es similar.

Lo que cambia esta vez es la vista del que se le muestra al cliente (figura. 5.14). Al igual que los trabajadores, el cliente también puede escanear el código QR pegado en el libro, pero a él solo se le mostrarán los datos relevantes de su préstamo y un botón para crear una incidencia. Para evitar que el cliente pueda cambiar la URL y acceder a vistas que él no podría ver, se ha creado el número de seguridad generado automáticamente.



Figura 5.14 : Vista del libro para el cliente

### 5.2.5 Gestión de accesos

Para controlar los accesos se dispone de una opción en el menú lateral (figura 5.6). Este menú está disponible en todo momento, por lo que el usuario podrá registrar un acceso en cualquier momento, sin importar de la acción que esté realizando. Como se puede apreciar en la figura 5.15, se disponen de 4 botones distintos: dos para registrar la entrada de hombre y mujer, y los otros dos para la salida de ambos sexos. Además se proporciona el valor total de personas que están dentro de la biblioteca en el momento actual. Es necesario diferenciar por sexo ya que esto será de mucha utilidad para la generación de estadísticas.



Figura 5.15 : Control accesos

## 5.2.6 Generación de estadísticas

Ya para acabar con toda la implementación, dejamos para la parte final las estadísticas, ya que se necesitaban todos los apartados previos para poder generarlas. Se pueden generar estadísticas tanto de accesos como de préstamos. Para ello, se ha creado una vista que separa estos dos tipos en *tabs* distintos dentro de los cuales están las estadísticas que se pueden generar sobre cada uno de ellos.

Para los accesos se pueden crear 4 tipos de estadísticas distintas, tal y como se puede apreciar en la figura 5.16: accesos por mes, accesos por año, accesos por día y accesos por franja horaria. Todos estas lanzan una vista similar a la figura 5.17, donde se muestra un gráfico de líneas que diferencia entre los accesos de hombres y mujeres, y una circular que muestra el total de accesos para cada sexo. Estas gráficas han sido creadas con Apache [31], que proporciona unas plantillas gratis que se pueden modificar para generar lo que se desee.

Por otro lado, existen dos tipos de estadísticas distintas que se pueden generar para los préstamos (figura 5.18). La primera de ellas, al igual que para los accesos, son gráficas que distinguen la cantidad de libros prestados para hombres y mujeres en un mes o en un año. El otro tipo de estadísticas son rankings de libros que más veces han sido prestados por mes, por año o en total. Este tipo lanza una vista similar a la figura 5.19, que es una tabla con el ranking solicitado. Se proporciona en formato PDF para que el usuario pueda imprimirlo.

The screenshot shows a web interface titled "Estadísticas". It has two main tabs: "Accesos" (selected) and "Préstamos". Under the "Accesos" tab, there are four sections for generating statistics:

- Accesos por mes:** Input fields for "Mes:" (5) and "Año:" (2021), with a "Sacar estadísticas" button.
- Accesos por año:** Input field for "Año:" (2021), with a "Sacar estadísticas" button.
- Accesos por día:** Input fields for "Día:" (17), "Mes:" (5), and "Año:" (2021), with a "Sacar estadísticas" button.
- Accesos por franja horaria:** Input fields for "Mes:" (5) and "Año:" (2021), with a "Sacar estadísticas" button.

Figura 5.16 : Estadísticas de accesos

## Estadísticas Accesos

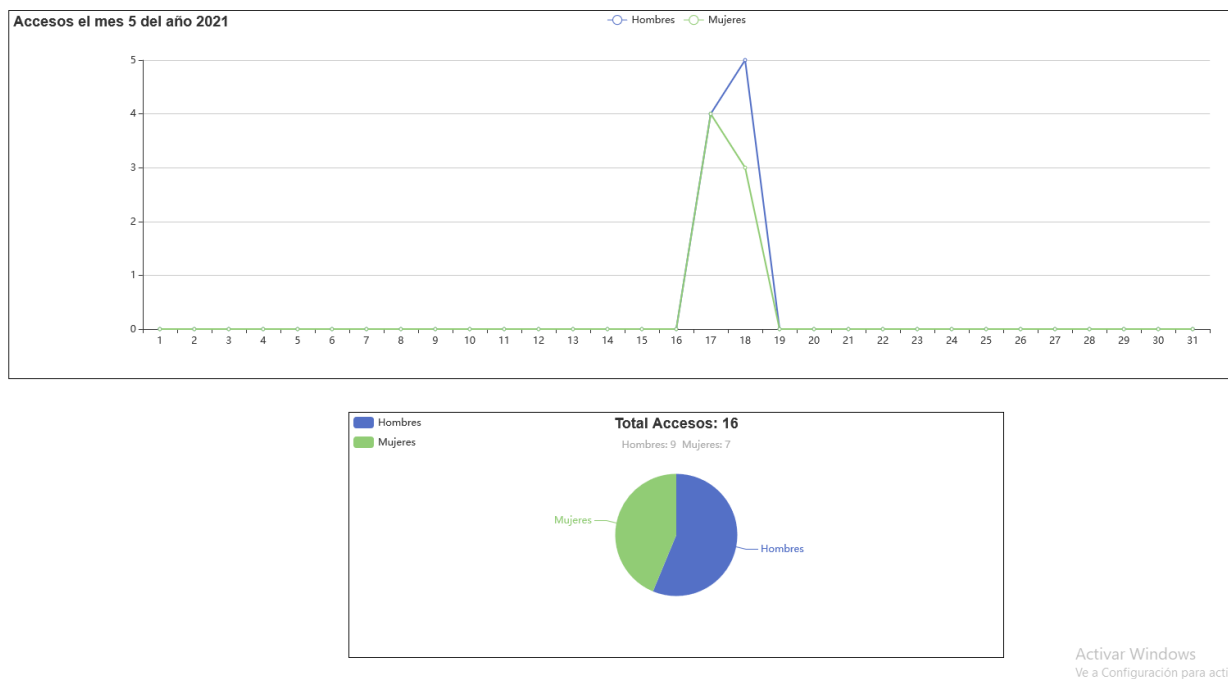


Figura 5.17 : Gráficas generadas

**Estadísticas**

**Accesos** | **Préstamos**

**Préstamos por mes:**

Mes:

Año:

**Préstamos por año:**

Año:

**Libros más prestados:**

Ranking de  libros

**Libros más prestados por mes:**

Ranking de  libros

Mes:

Año:

**Libros más prestados por año:**

Ranking de  libros

Año:

Figura 5.18 : Estadísticas de préstamos



## Libros más prestados en Mayo del 2021

Posición	Libro	Total veces prestado
1°	A algunos les gustan frías	2
2°	A aristocracia imposible	1
3°	-	-
4°	-	-
5°	-	-

Figura 5.19 : Ranking



## Capítulo 6

# Pruebas y validación

### 6.1 Pruebas unitarias y de integración

Una vez acabada la parte de implementación es necesario comprobar que todo funciona correctamente antes de dar el paso de ponerlo en marcha. Por ello es necesario hacer tanto pruebas unitarias como pruebas de integración, para que en caso de que se descubra cualquier fallo, poder solucionarlo rápidamente y así no entregar al cliente una versión defectuosa de la aplicación.

En nuestro caso, ya se fueron comprobando todos los resultados a medida que se realizaba cada implementación. Sin embargo, una vez esta fase acabó, se decidió hacer unas últimas pruebas a modo de validación. Primero se realizaron las pruebas unitarias de aquellas secciones que no dependían de otras para ser crear o para realizar alguna opción. Algunas de estas acciones son por ejemplo, la creación de libros y clientes o el control de accesos.

Tras esto, se hicieron las pruebas de integración para comprobar que las relaciones entre las distintas secciones funcionaban correctamente y sin fallos. Algunas de las funcionalidades que se comprobaron aquí son la creación de préstamos e incidencias para un libro realizadas tanto por un trabajador como por un cliente haciendo uso del código QR o la generación de estadísticas de accesos y de préstamos.

Tanto las pruebas unitarias como las de integración se hicieron manualmente, introduciendo datos falsos que permitieran realizar todas las comprobaciones necesarias.

### 6.2 Validación

Una vez realizadas todas las pruebas necesarias para verificar que todo lo desarrollado funciona correctamente, hay que revisar que se cumplen con todos los objetivos y requisitos establecidos en la planificación inicial, y que los diseños siguen la estructura propuesta en la sección de diseño.

Comenzando primero con los requisitos, es necesario comprobar que se cumplen todas los casos de uso identificados en el apartado 3.1. En nuestro caso, todas las

funcionalidades que había que programar se encuentran implementadas y BiblioCloud proporciona acciones para llevarlas a cabo.

En cuanto a la validación de las interfaces, tras comprobar exhaustivamente una por una, se verificó que cada vista cumplía con los requisitos establecidos en el apartado 4.2, es decir, los botones son iguales en todos los casos, el color principal es el mismo en toda la aplicación, las interfaces resultan agradables a la vista y son intuitivas. Además, para evitar que surja algún problema de comprensión, en las zonas donde se podía originar la confusión se han añadido textos de ayuda para evitar estos posibles incidentes.

Por último, es necesario comprobar que no hay ningún problema de seguridad. En nuestro caso, como ya se ha comentado en la parte de implementación, para evitar que el cliente que realiza el préstamo pueda acceder a vistas que no debe ver, se ha creado un código de seguridad el cual se ha añadido a la URL y que resulta prácticamente indescifrable para aquellos que no pertenezcan a la empresa Inttegrum.

## Capítulo 7

# Conclusiones

### 7.1 Conclusiones técnicas

El proyecto ha sido finalizado con éxito, consiguiendo desarrollar todas las funcionalidades que estaban previstas e incluso dándoles un aspecto mejor del que se esperaba. Además, se ha podido completar en el tiempo especificado al inicio del mismo.

Las tecnologías empleadas también han jugado un papel importante en el proyecto. Al ser fáciles de usar y disponer de una gran funcionalidad, han facilitado la implementación. Además, cuando surgía algún error o no nos acordábamos de cómo programar alguna parte, siempre podíamos consultar guías o foros en Internet que nos ayudaron a solucionar el problema, lo cual supuso una ventaja añadida.

### 7.2 Conclusiones personales

A nivel personal, trabajar en este proyecto ha sido una experiencia muy enriquecedora, tanto en el ámbito formativo como en el ámbito profesional.

Respecto al ámbito formativo, he aprendido a usar herramientas que no había utilizado antes, como es el caso del *framework* Yii2 o trabajar utilizando un servidor proporcionado por Plesk. He aprendido muchísimo sobre cómo desarrollar aplicaciones web y cómo adecuarse a las peticiones de los clientes, dándoles facilidades para que la aplicación les resulte sencilla y cómoda. Todo esto es algo que ya sabía antes de empezar las prácticas gracias a la formación recibida durante la carrera universitaria, sin embargo, haberlo experimentado en primera persona ha servido para aprender cómo se trabaja con aplicaciones web para clientes específicos en el mundo laboral.

En cuanto al ámbito profesional, me ha servido para mejorar en distintos aspectos tanto laborales como personales. He aprendido a trabajar en equipo, comunicando los avances y ayudando a los otros miembros de la organización. Además, he podido experimentar cómo es trabajar en un proyecto informático para clientes reales y cómo tratar con ellos.

Esta experiencia me ha permitido conocer cómo es el mundo laboral. Por eso, estoy muy agradecido con los miembros de Inttegrum por haberme acogido y haberme dado la oportunidad de colaborar con ellos en este proyecto.

# Bibliografía

- [1] Inttegrum: *Inttegrum - Servicios Integrales* [consultado el 13 de junio de 2021].  
<https://inttegrum.com/>
- [2] Mediterranea Gestión: *Mediterranea Gestión* [consultado el 13 de junio de 2021].  
<https://mediterraneagestion.es/>
- [3] Oropesa del Mar: *CIM (Centro Integral de Mayores)* [consultado el 13 de junio de 2021].  
<https://www.oropesadelmarturismo.com/ver/720/CIM-Centro-Integral-de-Mayores-.html>
- [4] Leomerly Salismey: *Gestión de Proyectos PMBOK: ¿Cómo funciona? - ComparaSoftware* [consultado el 13 de junio de 2021].  
<https://blog.comparasoftware.com/gestion-de-proyectos-pmbok/>
- [5] Plesk: *Plesk - Innovative Hosting Control Panel* [consultado el 13 de junio de 2021].  
<https://www.plesk.com/>
- [6] Plesk: *Primeros pasos en Plesk* [consultado el 13 de junio de 2021].  
<https://docs.plesk.com/es-ES/obsidian/customer-guide/primeros-pasos-en-plesk.69259/>
- [7] YiiFramework: *Yii PHP Framework* [consultado el 13 de junio de 2021].  
<https://www.yiiframework.com/>
- [8] Atom: *Atom* [consultado el 13 Junio 2021].  
<https://atom.io/>
- [9] PuTTY: *Dowloas PuTTY - a free SSH and telnet client for Windows* [consultado el 13 de junio de 2021].  
<https://www.putty.org/>
- [10] FileZilla: *Documentation - Filezilla Wiki* [consultado el 13 de junio de 2021].  
<https://wiki.filezilla-project.org/Documentation>
- [11] Plesk: *Plesk Pricing Plans & Options. Plesk License Types* [consultado el 13 de junio de 2021].  
[https://www.plesk.com/pricing/?utm\\_campaign=brand-eu&utm\\_source=google&utm\\_medium=ppc&utm\\_term=plesk](https://www.plesk.com/pricing/?utm_campaign=brand-eu&utm_source=google&utm_medium=ppc&utm_term=plesk)

- [12] IONOS: *Servidores Cloud a partir de 5€ por mes | IONOS de 1&1* [consultado el 13 de junio de 2021].  
<https://www.ionos.es/cloud/servidores-cloud#packages>
- [13] Glassdoor: *Sueldo: Programador Júnior* [consultado el 13 de junio de 2021].  
[https://www.glassdoor.es/Sueldos/programador-junior-sueldo-SRCH\\_KO0,18.htm](https://www.glassdoor.es/Sueldos/programador-junior-sueldo-SRCH_KO0,18.htm)
- [14] Universidad de Alicante: *Modelo vista controlador (MVC). Servicio de informática ASP.NET* [consultado el 13 de junio de 2021].  
<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- [15] Seul Lee: *Design Patterns, 01, MVC - Hello Python* [consultado el 13 Junio 2021].  
<https://lee-seul.github.io/concept/design-patterns/2017/03/06/design-pattern-01-MVC.html>
- [16] Agencia del ISBN: *Agencia del ISBN* [consultado el 13 de junio de 2021].  
<https://www.agenciaisbn.es/web/index.php>
- [17] Dilve: *Dilve - Distribuidor de información del libro en venta* [consultado el 13 de junio de 2021].  
<https://web.dilve.es/>
- [18] Dilve: *Descarga inicial del catálogo completo* [consultado el 13 de junio de 2021].  
<https://web.dilve.es/herramientas/descarga-inicial/>
- [19] Agencia Tributaria: *Tablas auxiliares de domicilios: provincias, municipios, ...* [consultado el 13 de junio de 2021].  
[https://www.agenciatributaria.es/AEAT.internet/Inicio/Ayuda/Tablas\\_auxiliares\\_de\\_domicilios\\_\\_provincias\\_\\_municipios\\_\\_\\_\\_\\_/Tablas\\_auxiliares\\_de\\_domicilios\\_\\_provincias\\_\\_municipios\\_\\_\\_\\_\\_.shtml](https://www.agenciatributaria.es/AEAT.internet/Inicio/Ayuda/Tablas_auxiliares_de_domicilios__provincias__municipios_____/Tablas_auxiliares_de_domicilios__provincias__municipios_____.shtml)
- [20] Font Awesome: *Free Icons* [consultado el 13 de junio de 2021].  
<https://fontawesome.com/icons?d=gallery&p=2&m=free/>
- [21] Bootstrap: *Buttons* [consultado el 13 de junio de 2021].  
<https://getbootstrap.com/docs/4.0/components/buttons>
- [22] Krajee: *DynaGrid* [consultado el 13 Junio 2021].  
<https://demos.krajee.com/dynagrid>
- [23] Krajee: *Tabs X* [consultado el 13 Junio 2021].



<https://demos.krajee.com/tabs-x#usage>

[24]GitHub: *2amigos QrCode Library* [consultado el 13 de junio de 2021].  
<https://github.com/2amigos/qrcode-library/blob/master/docs/index.md>

[25]Herma: *Schablone\_40x40\_10108\_9642 1* [consultado el 13 de junio de 2021].  
[https://www.herma.com.es/fileadmin/Buero-Zuhause/downloads/Stanzvorlagen/9642\\_SV.pdf](https://www.herma.com.es/fileadmin/Buero-Zuhause/downloads/Stanzvorlagen/9642_SV.pdf)

[26]Herma: *Etiquetas código, A4, 9642* [consultado el 13 de junio de 2021].  
<https://www.herma.com.es/oficina-hogar/producto/etiquetas-codigo-qr-9642/>

[27]Krajee: *mPDF* [consultado el 13 de junio de 2021].  
<https://demos.krajee.com/mpdf>

[28]Krajee: *Select2* [consultado el 13 de junio de 2021].  
<https://demos.krajee.com/widget-details/select2>

[29]Krajee: *Date Control* [consultado el 13 de junio de 2021].  
<https://demos.krajee.com/datecontrol>

[30]Dilve: *API de llamadas HTTP* [consultado el 13 de junio de 2021].  
<https://web.dilve.es/herramientas/api-de-llamadas-http/>

[31]Apache: *Examples - Apache Charts* [consultado el 13 de junio de 2021].  
<https://echarts.apache.org/examples/en/index.html>